

TRABAJO FINAL DE GRADO

Grado en Ingeniería Electrónica Industrial y Automática

SUPERVISIÓN Y CONTROL DE UN SISTEMA DE ENSABLAJE FLEXIBLE



Memoria y Apéndices

Autor: Adrian Sebastian Minea
Director: Javier Gámiz Caro
Departamento: ESAll
Convocatoria: Enero 2020

Resum

L'actual projecte es centra en la automatitzaci3n del sistema FAS200 present al laboratori de automàtica y rob3tica de l'Escola d'Enginyeria de Barcelona Est (EEBE). Aquest sistema esta format per 6 cèl·lules que poden ser controlades independentment pels seus controladors *MicroLogix*. L'objectiu del projecte es poder controlar aquest sistema mitjançant un sol controlador (*CompactLogix*) y centralitzar el control i la supervisi3n en un sol equip, mitjançant la implementaci3n d'un sistema SCADA.

Resumen

El actual proyecto se centra en la automatizaci3n del sistema FAS200 presente en el laboratorio de autom1tica y rob3tica de la *Escola d'Enginyeria de Barcelona Est* (EEBE). Este sistema est1 compuesto por 6 c3lulas que pueden ser controladas independientemente por sus controladores *MicroLogix*. El objetivo del proyecto es poder controlar dicho sistema mediante un solo controlador (*CompactLogix*) y centralizar el control y la supervisi3n en un solo equipo, mediante la implementaci3n de un sistema SCADA.

Abstract

The present project focuses on the automation of the FAS200 system installed in the automation and robotics laboratory of the Escola d'Enginyeria de Barcelona Est (EEBE). This system is composed of 6 cells that can be independently controlled by their MicroLogix controllers. The purpose of the project is to be able to control this system by means of a single controller (*CompactLogix*) and to centralize the control and supervision in a single equipment, through the implementation of a SCADA system.

Agradecimientos

Este proyecto no se podría haber llevado a cabo sin la ayuda de un gran número de personas.

En primer lugar, mi tutor, Javier Gámiz, le debo dar las gracias por la propuesta que me hizo, le debo gran parte de los conocimientos que he adquirido estos últimos meses.

Por otro lado, a toda mi familia, por siempre apoyarme y siempre sacar lo mejor de mi mismo, hasta en los peores momentos.

Por último, agradecer a los maestros de laboratorio por dejarme tener acceso a las instalaciones del laboratorio y ayudarme siempre con cualquier duda, consulta o problema.

Índice

1.	Introducci3n.....	1
1.1.	Origen y Motivaci3n del Proyecto.....	1
1.2.	Objetivos	1
1.3.	Justificaci3n	2
1.4.	Alcance	2
1.5.	Estructura y Organizaci3n del Proyecto	3
2.	Análisis del Sistema	5
2.1.	Estudio del Proceso por Automatizar.....	5
2.2.	Sistema de Control.....	6
2.2.1.	FAS-230.....	6
2.2.2.	FAS-201.....	8
2.2.3.	FAS-202.....	11
2.2.4.	FAS-203.....	13
2.2.5.	FAS-204.....	16
2.2.6.	FAS-207.....	18
2.2.7.	FAS-208.....	22
2.3.	Requerimientos funcionales.....	25
2.4.	Requerimientos de diseño.....	26
2.5.	Metodología de desarrollo	28
2.6.	Planificaci3n de las tareas	29
3.	Diseño e implementaci3n del proyecto.....	31
3.1.	Arquitectura del sistema de control	31
3.1.1.	Hardware del sistema	31
3.1.2.	Software del sistema	32
3.1.3.	Comunicaciones SCADA - Controlador (PLC) - Proceso.....	33
3.2.	Descomposici3n del proyecto a automatizar	38
3.3.	Codificaci3n de los elementos y sistemas.....	39
3.4.	Definici3n de la interfaz del sistema de control	43
3.5.	Fichero de intercambio entre el controlador y SCADA	44
3.6.	Programa del controlador	48
3.6.1.	Estructura del controlador	48

3.6.2.	Definición del tipo de datos	50
3.6.3.	Lógica de control de los elementos del sistema	54
3.6.4.	Lógica de control de los sistemas.....	60
3.6.5.	Secuencias de control	63
3.7.	Programa de la aplicación SCADA	76
3.7.1.	Árbol de navegación de la aplicación.....	76
3.7.2.	Definición del tipo de datos y <i>tags</i>	77
3.7.3.	Scripts	78
3.7.4.	Diseño de las ventanas principales de la aplicación	79
3.7.5.	Diseño de la ventana de alarmas de la aplicación	86
3.7.6.	Ventana de registros del sistema	87
3.7.7.	Ventanas secundarias del sistema	87
4.	Pruebas y resultados	89
4.1.	Pruebas.....	89
4.2.	Resultados	89
5.	Normativa	104
5.1.	Codificación de los elementos.....	104
5.2.	Programación PLC	105
5.3.	Programación aplicación SCADA	105
6.	Conclusiones.....	106
7.	Bibliografía.....	107
ANEXOS	108

1. Introducci3n

La automatizaci3n industrial es un conjunto de m3todos y recursos, que implican equipos f3sicos, conocidos como el *hardware* del sistema y los programas, conocidos con *software*, que definir3n el comportamiento de los equipos f3sicos. Mediante la l3gica programada y la electr3nica, somos capaces de poder dar3r3rdenes a este *hardware* y hacer que se comporte de la forma que nosotros deseemos.

1.1. Origen y Motivaci3n del Proyecto

El tema principal de ingenier3a alrededor del cual gira todo el proyecto es la automatizaci3n industrial, teniendo esto en cuenta, se debe de considerar la importancia que ya tiene dicho concepto en la industria y el gran crecimiento que est3 teniendo actualmente.

La implementaci3n de estos sistemas automatizados dentro de la industria hace muchos procesos pasen a ser mucho m3s precisos, r3pidos, eficientes, en definitiva, con menores3ndices de desperdicio y la posibilidad de programar la producci3n de un determinado producto durante las 24 horas del d3a, los siete d3as de la semana, siempre supervisado por un usuario humano.

Por otro lado, tambi3n nos ayuda a poder tener un mayor control sobre el sistema, posibles malas pr3cticas, alarmas o incidentes. Tambi3n a poder almacenar mejor las estad3sticas productivas del proceso lo cual nos aporta informaci3n muy valiosa para poder optimizar las necesidades del sistema y de los usuarios.

Por lo tanto, como la intenci3n principal de este proyecto es profundizar en los aspectos anteriormente comentados, aporta un nivel de aprendizaje muy3til y totalmente necesario de cara a posibles salidas laborales dentro del sector. Es un paso inicial para poder comenzar a tratar la metodolog3a, los pasos, los posibles problemas y consecuentes soluciones que un proyecto de esta3ndole puede originar.

1.2. Objetivos

El objetivo principal del proyecto es el dise3o y la implementaci3n de un sistema de control y supervisi3n. El control del sistema se consigue mediante un dispositivo l3gico programable (PLC) y la supervisi3n gracias a un sistema de supervisi3n y adquisici3n de datos SCADA (Supervisory Control And Data Adquisition) sobre las c3lulas automatizables del laboratorio de rob3tica de la EEBE (*Escola d'Enginyeria de Barcelona Est*).

El aut3mata programable que se ha comentado anteriormente va a funcionar como *front-end*, ya que las diferentes c3lulas automatizables del sistema constan, cada una, con una unidad de control. Este aut3mata principal va a gestionar todas las se3ales del sistema procedente de las diferentes c3lulas. Uno de los objetivos es que el PLC sea capaz de poder controlar y llevar a cabo toda la mensajer3a entre3l y los controladores de cada c3lula.

Por otro lado, otro de los objetivos del proyecto, como se ha comentado, es la supervisi3n remota del sistema. Esto permite tener un acceso m3s eficiente a las posibles3rdenes que le

podemos dar al sistema, visualizar los estados del sistema, acceso a información relevante y posibles problemas y/o alarmas que se pueden generar.

El logro de estos objetivos principales viene ligado a poder utilizar el material creado en este proyecto como herramienta de docencia para cursos posteriores, ya que, para el desarrollo del proyecto, se seguirá la metodología y se pondrán en práctica los conocimientos adquiridos durante estos años en la escuela. Esto pone de manifiesto el objetivo de querer ampliar y utilizar los conocimientos adquiridos para poder desarrollar un proyecto de esta envergadura y ser capaz de utilizar de manera eficiente dichos conocimientos.

1.3. Justificación

El actual proyecto se centra en el sistema de células automatizables (FAS-200) proporcionadas por SMC que se encuentran en el laboratorio de robótica (A5.4).

La FAS-200 es un sistema que puede estar compuesto por tanto módulos como deseemos, en este caso, son seis (FAS 201, 202, 203, 204, 207 y 208). Por último, hay otro modulo (FAS-230) que es el que sirve de comunicador/transportador de piezas entre los diferentes módulos. Estos módulos vienen configurados de fábrica para poder ser controlados de manera independiente pero su uso efectivo es cuando se utiliza, al menos, con su pareja inmediata, es decir, 201 junto con 202, 203 junto con 204 y por último 207 con 208.

De tal manera, aún que cada módulo se pueda controlar de manera independiente es necesario utilizar un PLC *front-end* para poder controlar simultáneamente cada pareja de módulos y hacer que su comportamiento sea más eficiente en nuestro sistema. La eficiencia aumenta en utilizar un *CompactLogix*, el motivo es que el entorno de programación del PLC *Front-End* es mucho más potente que el del *MicroLogix*, ya que el *CompactLogix* puede ser programado en múltiples lenguajes de programación (*ladder*, *SFC*, *FB*, etc) mientras que el *MicroLogix* solo puede ser programado en *ladder*. Además, en el *CompactLogix* se pueden crear *tags* propios, mientras que en el *MicroLogix* viene ya prefijados.

Como el objetivo de la universidad es utilizar este sistema en una vertiente docente, para poder exponer a los estudiantes de una manera práctica los conceptos a adquirir durante el curso, es de obligado cumplimiento utilizar los conocimientos adquiridos anteriormente para poder automatizar de manera eficiente el sistema en cuestión.

1.4. Alcance

El proyecto comienza por hacer un estudio del funcionamiento del proceso a automatizar, de todas las variables que componen el sistema y tener clara su función. La posterior codificación, primero de los registros presente en los controladores *MicroLogix* para crear una nomenclatura de enlace entre estos registros de entradas y salidas reales con los que llegan finalmente al PLC *front-end* y por último la nomenclatura de todos los *tags* del programa del controlador principal que posteriormente va a definir el fichero de intercambio entre el controlador y el SCADA.

Seguidamente y con las señales debidamente definidas se procede a realizar la programación del PLC *front-end*, con diferentes formas de funcionamiento, manual, secuencia automática,

estado de mantenimiento, condiciones iniciales del sistema general o por partes, con capacidad de poder controlar todo el sistema. Sacándole el máximo partido a todos los sensores del sistema que nos permite tener un control sobre el producto final, pudiendo elegir la fisonomía y anatomía del producto.

Por otro lado, como hemos comentado se realiza la creación de una aplicación SCADA, mediante la cual se va a poder controlar el sistema de manera remota, teniendo control total sobre el sistema en todos los modos de funcionamiento. Por otro lado, podremos obtener la información de manera más eficiente, rápida, segura y mejor organizada. La plataforma destinada al diseño SCADA es InTouch de Wonderware, el cual nos va a permitir la supervisión y control del sistema.

Todo los programas y aplicaciones creados cumplen con los requisitos de funcionamiento y diseño que están estipulados en el presente proyecto en el apartado 2.

1.5. Estructura y Organización del Proyecto

Se ha pensado y dividido el actual proyecto en 8 partes diferenciadas en las que podemos ver la evolución que se va logrando, desde la exposición del proceso a automatizar y programar, hasta el apartado de pruebas finales y las conclusiones del proyecto realizado.

- ❖ **Introducción:** se exponen los objetivos, el motivo y el alcance del proyecto realizado.
- ❖ **Análisis del sistema:** se expone el sistema con el que se va a tratar, sus diferentes partes, detalle de los procesos que intervienen, su funcionamiento general, los requerimientos demandados que se deben llevar a cabo para crear un sistema eficiente y las características de cada subsistema, dando una vista general de todo lo que debemos implementar en el sistema.
- ❖ **Solución propuesta al sistema. Diseño e implementación:** apartado en el cual se especifica la solución propuesta al sistema. Se justifica la programación creada para controlar el autómatas, la creación de la aplicación SCADA, la codificación de los elementos del sistema, los ficheros de intercambio, etc. En definitiva, se exponen todas las decisiones tomadas para la creación del proyecto.
- ❖ **Pruebas y resultados:** se llevan a cabo todas las pruebas, tanto FAT como SAT, para comprobar el correcto funcionamiento del sistema.
- ❖ **Normativa del proyecto:** se expone toda la normativa que se ha seguido en todos los ámbitos para la creación del proyecto, tanto en la implementación como en el diseño.
- ❖ **Conclusiones:** valoración final del proyecto y de todo su desarrollo, se comenta posibles detalles que pueden haber influido a su resultado final.

- ❖ **Bibliografía:** apartado para especificar las fuentes de informaci3n que se han utilizado en la redacci3n del proyecto y que han servido de apoyo, tanto en la redacci3n como en la programaci3n y parte pr3ctica del proyecto.

Por otro lado, se expone un apartado de ap3ndices donde podemos observar toda la programaci3n creada y por 3ltimo un presupuesto del proyecto, lo m3s cercano posible a la realidad.

2. Análisis del Sistema

En el actual apartado se procede a estudiar el funcionamiento del sistema y/o subsistemas, los componentes y especificaciones técnicas del proceso sobre el cual se ha llevado a cabo el actual proyecto de automatización. En función del actual análisis, que incluye las especificaciones y requerimiento funcionales y de diseño del sistema y de la aplicación SCADA, se va a desarrollar la solución al objetivo de automatización, siempre cumpliendo con dichas especificaciones y requerimientos previamente establecidos.

2.1. Estudio del Proceso por Automatizar

El proceso que se debe automatizar, como se ha comentado en la introducción del proyecto es el sistema FAS-200 presente en el laboratorio de robótica.

Este sistema está formado por diferentes módulos, 17 en concreto (podemos ver una disposición general en la figura 2.1.), en nuestro caso está formado por seis módulos, dichos módulos pueden operar de manera independiente, ya que cada uno de ellos están dotados de un PLC, en nuestro caso del fabricante Allen Bradley. Pero es cuando se agrupan por parejas o en grupos más grandes cuando son más eficientes y se les pueden sacar el mayor partido.

El objetivo final es poder controlar y supervisar el funcionamiento de los seis módulos simultáneamente, para crear una producción automatizada del sistema y controlarlo de manera óptima.

Estos seis módulos, con sus seis PLC's diferentes, modelo MicroLogix 1766-L32BXBA se comunican mediante Ethernet IP con el PLC *Front-End* que controla todo el funcionamiento y comportamiento del sistema.

Los diferentes PLC's que se encuentran en las células de ensamblaje, sirven de *GateWay* entre los equipos físicos y el PLC principal, modelo CompactLogix 1769-L16ER-BB1B, que gobierna el funcionamiento principal del sistema.

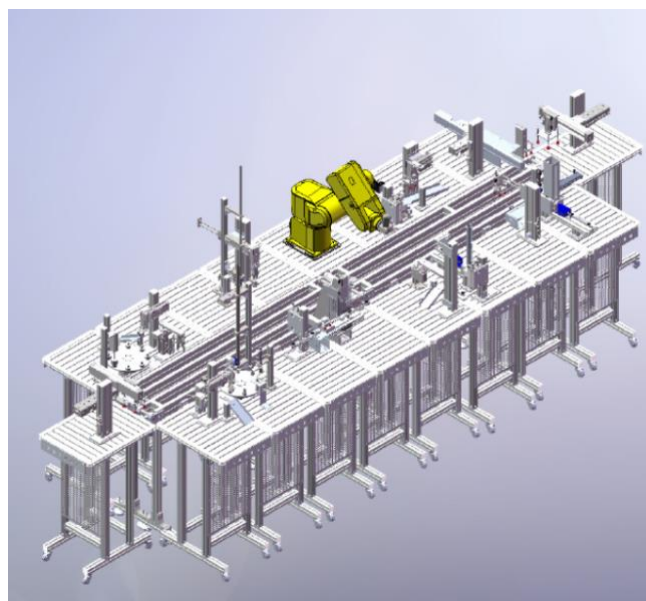


Fig.- 2.1. Disposición general de todos los módulos de la FAS-200. Fuente: (www.google.es).

2.2. Sistema de Control

En nuestro caso disponemos de 6 de los módulos del fabricante. Funcionan por parejas y se deben combinar para obtener el producto final, un mecanismo de giro.

Es un sistema de ensamblaje flexible ya que nos permite tener impacto sobre él y dependiendo de las ordenes que reciba del usuario puede llegar a ofrecer distintas variaciones del producto final. Se puede variar el material del producto y las dimensiones, el sistema tiene la capacidad para poder diferenciar entre un tipo u otro de material y de dimensión lo que lo hace un buen aliado para poder crear productos de diferente composición y apariencia.

Los módulos que están presentes en el laboratorio y se han automatizado son los siguientes, se disponen por parejas:

- ❖ FAS-201 (estación de alimentación y verificación de la orientación de la base)
- ❖ FAS-202 (estación de rechazo/transferencia de la base)

- ❖ FAS-203 (estación de alimentación y transferencia de rodamientos)
- ❖ FAS-204 (estación de medida y transferencia de rodamientos)

- ❖ FAS-207 (estación de alimentación y clasificación de ejes)
- ❖ FAS-208 (estación de rechazo/transferencia de ejes)

Además de estos módulos, se dispone de la estación FAS-230, este módulo es un transfer que transporta el producto en proceso a lo largo de todas las estaciones del sistema. Cada transfer conecta 4 estaciones, pero podemos disponer de la opción de enlace entre 6 estaciones, como la que se encuentra en el sistema en cuestión.

En los siguientes apartados se dispone a exponer las diferentes estaciones de una manera más extensa, explicando todos los subsistemas que los componen para entender en profundidad y más exactamente su funcionamiento, para la posterior solución de automatización que se lleva a cabo.

2.2.1. FAS-230

Este módulo sirve de transporte para las piezas del sistema, dependiendo del objetivo del sistema, puede tener diferentes dimensiones en función del número de módulo que queramos incorporar. En nuestro caso, contamos con un transfer que une 6 módulos.



Fig.- 2.2. Transfer FAS-230. Fuente: (www.smc.eu).

El transfer incorpora unos *palets* de para transportar las piezas y montajes entre las diferentes estaciones de manipulación. Los *palets*, como podemos ver en la figura 2.3., junto con los

sensores inductivos de la cinta conforman una codificación binaria, que permiten al sistema de control identificar la posición de cada *palet* en cualquier instante.

Dependiendo de los módulos que tengamos incorporados, el transfer tendrá diferentes mecanismos delante de cada estación, como se puede observar en la figura 2.4., son una serie de manipuladores para elevar, centrar y girar el producto que está siendo ensamblado y controlar el comportamiento de los *stoppers*, para dejar avanzar o detener el producto mientras esta siendo manipulado por la estación en curso.



Fig.- 2.3. Palets con código binario del transfer FAS-230. Fuente: (www.smc.eu)



Fig.- 2.4. Mecanismos presentes en la FAS-230 (*stoppers*, elevador con microrruptor y detectores inductivos). Fuente: (www.smc.eu).

DATOS TÉCNICOS (FAS-230):

Dimensiones para 6 módulos: 1 sección 2700 x 320 mm, altura 940mm.

1 motor de 24 VDC HIWIN (AM1 M12029A1).

1 Fuente de alimentación 24 VDC/60 W OMRON (S8VK-G06024).

Actuadores

1 cilindro compacto de doble efecto Ø32, carrera: 25 mm (CDQMB32-25). Controlado por una electroválvula monoestable 5/2. (cada 2 módulos).

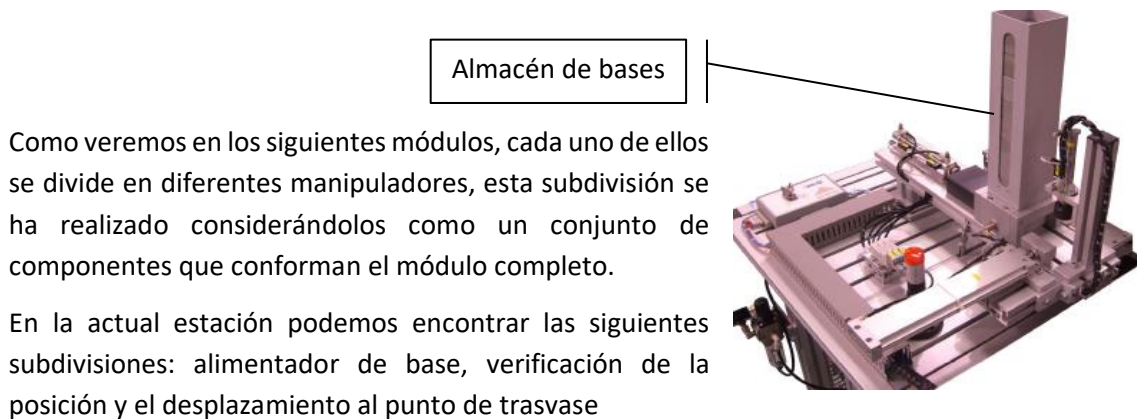
1 cilindro compacto de doble efecto con guías Ø16, carrera: 30 mm (MGPM16-30Z), con reguladores de caudal. Controlado por una electroválvula monoestable 5/2. (algunos módulos).
1 Mesa giratoria de dos direcciones θ máx.: 90° (MSUB3-90S), con reguladores de caudal. Controlado por una electroválvula monoestable 5/2. (algunos módulos).

Sensores

3 detectores inductivos OMRON (E2A-M12KS04-M1-B1). (cada 2 módulos).
1 microrruptor OMRON (V-166-1C5). (cada 2 módulos).

2.2.2. FAS-201

Este primer módulo es el que inicia el sistema de ensamblaje, tiene la función de alimentar el cuerpo o base que va a cumplir la función de soporte del mecanismo de giro, verifica su orientación y correcto posicionamiento y lo traslada a la posición final (localizada en el siguiente módulo FAS-202) para seguir con el ensamblaje del producto.



Como veremos en los siguientes módulos, cada uno de ellos se divide en diferentes manipuladores, esta subdivisión se ha realizado considerándolos como un conjunto de componentes que conforman el módulo completo.

En la actual estación podemos encontrar las siguientes subdivisiones: alimentador de base, verificación de la posición y el desplazamiento al punto de trasvase

Fig.- 2.5. FAS-201. Fuente: (www.smc.eu).

Alimentación de la base

La incorporación de las bases al sistema se produce mediante un alimentador por gravedad, las bases permanecen almacenadas unas encima de otras, (figura 2.5.) de forma que al alimentar la situada en la parte inferior el resto caen por su propio peso.

La operación de extraer la base del alimentador se realiza mediante un cilindro neumático, que impulsa un empujador de forma similar al perfil de la base, que lo emplaza en la siguiente fase, la verificación de la posición.

También se incorpora un sensor inductivo para poder comprobar la ausencia de bases. En la figura 2.6. podemos ver los componentes de este subsistema.



Fig.- 2.6. Alimentación de la base. Fuente: (www.smc.eu).

Verificación de la base

La base, que sirve como alojamiento del resto de los componentes que se van ensamblando para producir el producto final, debe tener una orientación determinada para su correcta utilización y emplazamiento en el *palet* que lo transporta por lo largo de todo el proceso.

Para ello, el sistema se sirve de un cilindro que se introduce en el orificio de la base para comprobar la orientación. Si la base se encuentra invertida, con la orientación equivocada, el cilindro no llegará a efectuar su completa carrera y el sensor magnético que posee no se activa, esta señal se introduce al PLC que sirve para tener conocimiento de si la base se encuentra en posición correcta o no. Si la base esta situada correctamente, el cilindro llega a su final de carrera, por lo tanto, obteniendo esta señal podemos verificar la correcta orientación de la base.

En la figura 2.7. se puede observar el cilindro junto al objeto en su extremo, que sirve para la comprobación de la orientación.

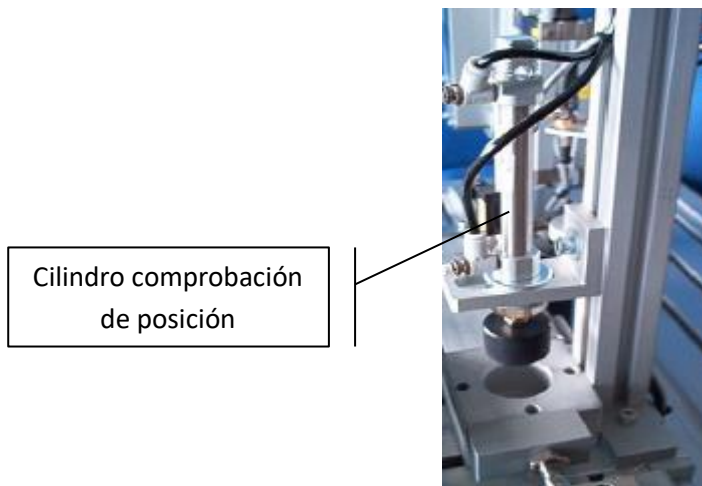


Fig.- 2.7. Verificación de la orientación de la base. Fuente: (www.smc.eu).

Desplazamiento al punto de trasvase

Para trasladar la base al siguiente módulo se utiliza otro cilindro neumático con un empujador en su extremo, este tiene una sección rectangular para no afectar la orientación de la base antes

de su trasvase al *palet* presente en la cinta transportadora (FAS-230), proceso que se lleva a cabo en la siguiente estación (FAS-202).

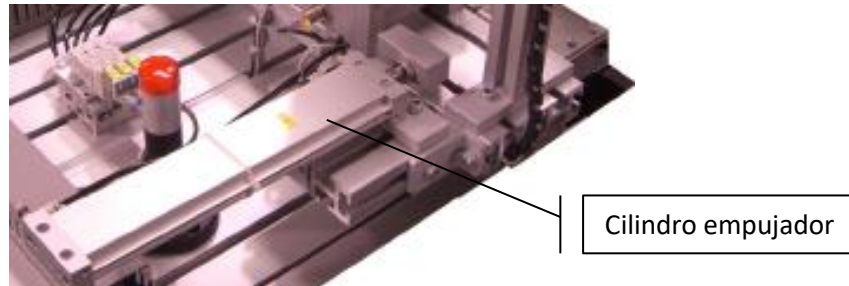


Fig.- 2.8. Desplazador al siguiente módulo. Fuente: (www.smc.eu).

DATOS TÉCNICOS (FAS-201):

Alimentador de base

Con capacidad de almacenaje de hasta 12 bases dentro del suministrador por gravedad.

Actuadores

1 cilindro empujador de doble efecto $\varnothing 16$, carrera: 100 mm (CD85N16-100-B), con reguladores de caudal y detectores de posición inicial y final. Controlando por una electroválvula monoestable 5/2.

Sensores

2 detectores magnéticos Reed (D-C73L).

1 detector inductivo OMRON (E2A-M12KS04-WP-B1).

Verificación de la base

Actuadores

1 cilindro de doble efecto $\varnothing 12$, carrera: 50 mm (CD85N12-50-A), con reguladores de caudal y detector de posición final. Controlado por una electroválvula monoestable 5/2.

Sensores

1 detector magnético tipo Reed (D-A73CL).

Desplazamiento al punto de trasvase

Actuadores

1 cilindro empujador de sección rectangular $\varnothing 25$, carrera: 200 mm (MDUB25-200DMZ), con reguladores de caudal y detector de posición final. Controlado por una electroválvula monoestable 5/2.

Sensores

1 detector magnético tipo Reed (D-A93L).

2.2.3. FAS-202

Este segundo módulo es complementario al anterior, una vez la FAS-201 incorpora en esta segunda estación la base, nuestro PLC busca el *tag* de comprobación de la orientación. Dicho *tag* ha sido modificado por la FAS-201 en el momento de saber si la orientación de la base es correcta o no, dependiendo del estado de este *tag* la FAS-202, rechazará la base o la mantendrá en el sistema.

Si el *tag* se encuentra a '1' el sistema se encarga de recoger la base y trasladarla encima del *palet* para ser transportada a la siguiente estación de ensamblaje. El sistema empleado es explicado en los siguientes apartados.

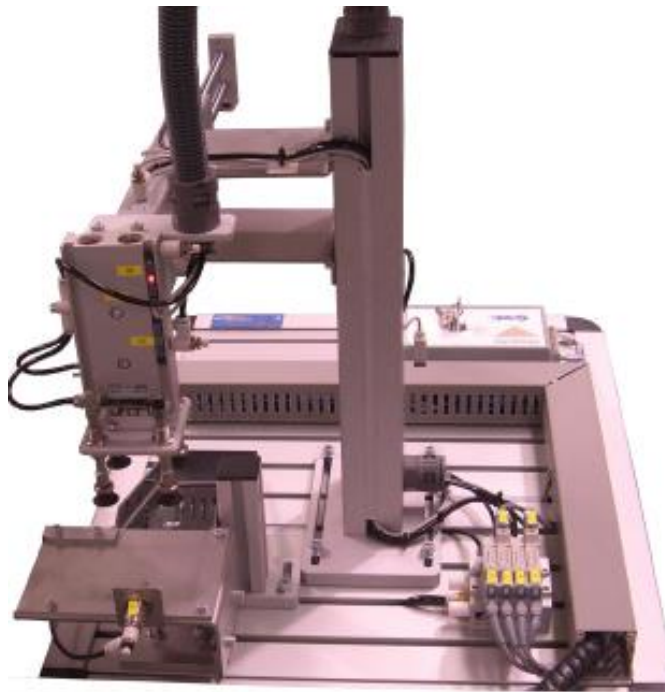


Fig.- 2.9. FAS-202. Fuente: (www.smc.eu).

En esta estación de ensamblaje podemos encontrar las siguientes subdivisiones: rechazo de base incorrecta e inserción de la base en *palet*.

Rechazo de la base incorrecta

La acción de esta parte del módulo va en función de si la orientación de la base es correcta, si no es así el cilindro de simple efecto se activa e impulsa la base directamente a la zona de rechazo, dejando la zona de inserción libre para situar una nueva base. En la figura 2.10. podemos observar el cilindro que se encarga de rechazar las piezas erróneas.

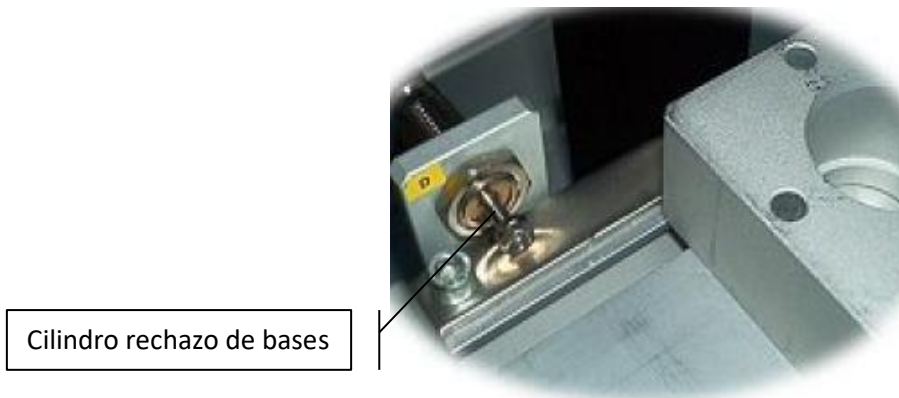


Fig.- 2.10. Rechazo de bases erróneas. Fuente: (www.smc.eu).

Inserción de base en el *palet*

Después de comprobar la buena orientación de la base, se procede a insertarla en el *palet* que se encuentra justo delante de la estación. Esta tarea se lleva a cabo mediante un manipulador cartesiano de dos ejes, cada eje está constituido por un cilindro neumático de vástagos paralelos, para poder moverlo en el eje vertical y en el horizontal.

Como elemento terminal posee una plataforma de sujeción por vacío, que incorpora 4 ventosas telescópicas con el fin de absorber posibles desalineaciones en altura. En la figura 2.11. podemos apreciar el manipulador de dos ejes junto al sistema de vacío.

La aspiración de aire se logra mediante un eyector de vacío, junto con un vacuostato que proporciona al PLC una señal en el momento que la pieza esta sujeta de manera correcta, cuando el vacío creado es el adecuado para poder elevar la pieza y para asegurar la correcta sujeción, dicho sensor se puede observar el sistema en la figura 2.12.

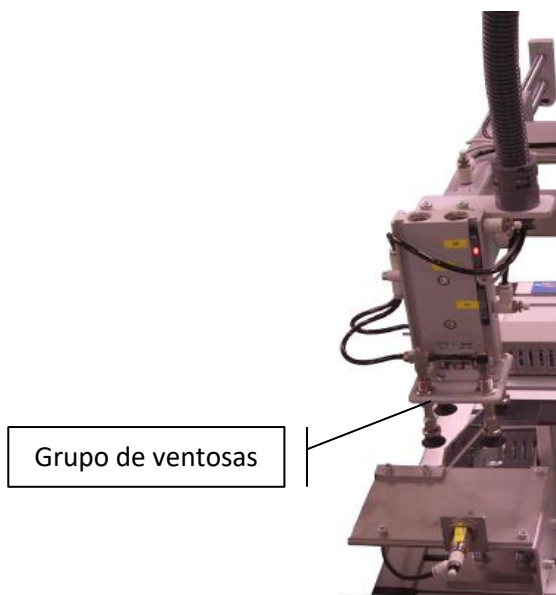


Fig.- 2.11. Manipulador. Fuente: (www.smc.eu)

Fig.- 2.12. Sistema de vacío. Fuente: (www.smc.eu).

DATOS TÉCNICOS (FAS-202):

Rechazo de la base incorrecta

Actuadores

1 cilindro de rechazo de simple efecto Ø10 carrera: 15 mm (CJPB10-15H6). Controlado por una electroválvula monoestable 3/2.

Inserción de base en el palet

Actuadores

1 cilindro de doble vástago y vástagos paralelos (eje horizontal) Ø20, carrera: 150 mm (CXSWM20-150-XB11), con regulador de caudal y detectores de posición inicial y final. Controlado por una electroválvula biestable 5/2.

1 cilindro de vástagos paralelos (eje vertical) Ø15 carrera: 50 mm (CXSM15-50), con reguladores de caudal y detectores de posición inicial y final. Controlados por una electroválvula monoestable 5/2.

4 ventosas telescópicas Ø16 (ZPT16CNK10-B5-A10) con eyector para generar el vacío (ZU07S), para el sistema de sujeción por vacío. Controlado por una electroválvula biestable 3/2.

Sensores

4 detectores magnéticos tipo Reed (D-Z73L).

1 vacuostato con salida PNP (PS1100-R06L).

2.2.4. FAS-203

Esta estación es la encargada de introducir al sistema los rodamientos que van ensamblados en la base anteriormente tratada. Traslada el rodamiento a la siguiente estación (FAS-204), para ser ubicada junto a la base. También el actual módulo traslada el rodamiento a la posición de medición de los rodamientos, presente en el siguiente módulo, donde la altura es la medida para diferenciar entre los dos tamaños de rodamiento (alto y bajo). En la figura 2.13. podemos observar la FAS-203.

Al igual que los demás módulos, también se divide entre diferentes subdivisiones para facilitar su comprensión: alimentador del rodamiento y el trasvase a la posición de medición.

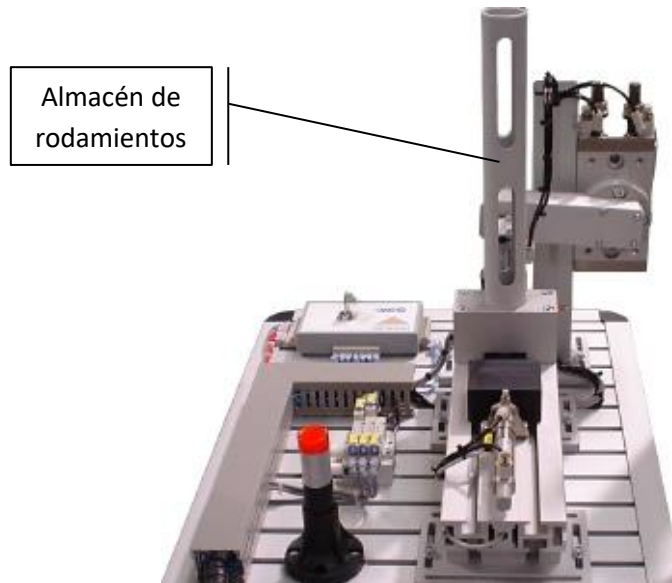


Fig.- 2.13. FAS-203. Fuente: (www.scm.eu).

Alimentador de rodamientos

Mediante el mismo mecanismo que se utiliza para el almacenamiento de las bases en la FAS-201, se almacenan los rodamientos antes de ser introducidos al sistema de ensamblaje. Es un alimentador por gravedad, que, al ser introducido el rodamiento inferior, los demás caen por su propio peso. Este rodamiento es introducido al sistema y trasladado a la siguiente posición mediante un cilindro neumático.

El sistema consta de un microrruptor que nos servirá de comprobación de la existencia de rodamiento o si por el contrario se han agotado.

En la figura 2.14. se observa el cilindro empujador de rodamientos y la estructura donde se encuentran los rodamientos listos para su entrada al circuito de ensamblaje.

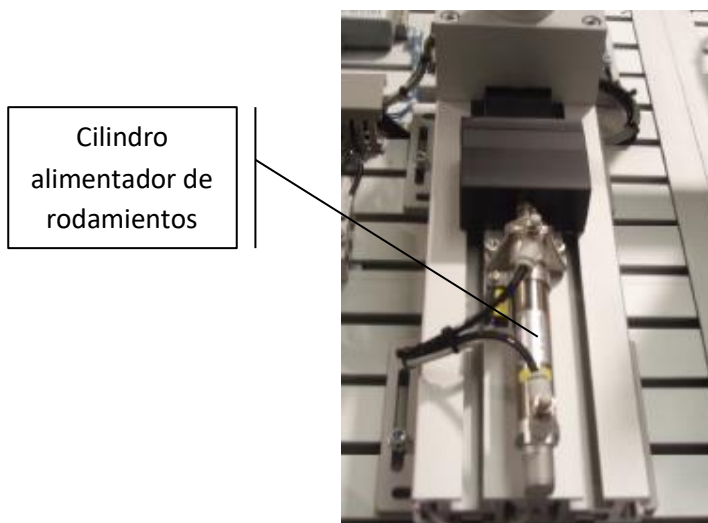


Fig.- 2.14. Alimentador de rodamientos. Fuente: (www.smc.eu).

Transvase a la posición de medición

Una vez alimentado el rodamiento y presente en el sistema y con la señal del microrruptor de que hay presencia, se procede a posicionar el rodamiento en la siguiente estación, donde va a ser tratado antes de ser posicionado junto a la base.

Para realizar el desplazamiento desde el punto de alimentación (FAS-203), al lugar donde se realiza la medición del rodamiento (FAS-204), se utiliza un manipulador con un mecanismo de giro piñón-cremallera, que describe un movimiento angular de 180°. En el eje del mecanismo de giro se ha acoplado un brazo el cual en su terminación posee una pinza neumática que sujeta el rodamiento por la parte interna en el momento del trasvase.

El brazo (figura 2.15.), en su interior, tiene un mecanismo de dos piñones y una correa dentada, que facilita que su terminación, donde se encuentra la pinza junto al rodamiento, no cambie de orientación durante su movimiento angular. De este modo podemos emplazar el rodamiento en el punto final sin ningún ángulo de inclinación.

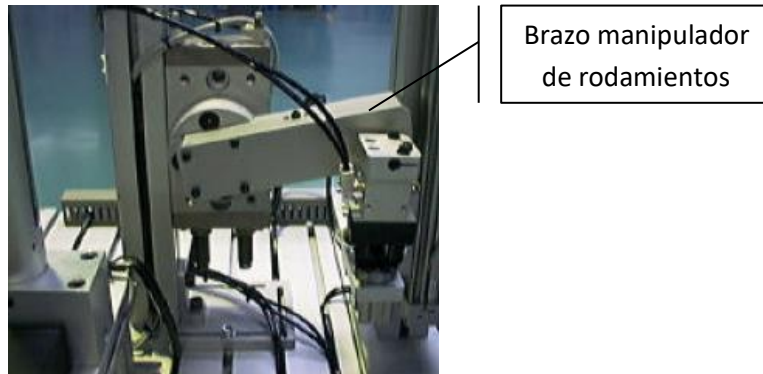


Fig.- 2.15. Manipulador de giro para el trasvase de los rodamientos. Fuente: (www.smc.eu).

DATOS TÉCNICOS (FAS-203):

Alimentador de rodamientos

Actuadores

1 cilindro empujador de doble efecto Ø16, carrera: 50 mm (CD85N16-50B), con reguladores de caudal y detector de posición final. Controlado por una electroválvula monoestable 5/2.

Sensores

1 detector magnético tipo Reed (D-C73L).

1 detector de presencia: microrruptor OMRON (V-166-1C5).

Transvase a la posición de medición

Actuadores

Dispositivo de giro: 1 actuador de giro tipo piñón-cremallera Ø50, $\Omega_{\text{máx.}}$: 180° (MSQB50A), con regulador de caudal y detectores de posición de 0°, 90° y 180°. Controlado por una electroválvula 5/3 de centros cerrados.

Brazo de sujeción: 1 pinza neumática de dos dedos de apertura paralela (MHK2-16D).
Controlada por una electroválvula biestable 5/2.

Sensores

3 detectores magnéticos tipo Reed (D-A93L).

2.2.5. FAS-204

Este cuarto módulo (figura 2.16.) es complementario del anterior y tiene la función de medir la altura del rodamiento suministrado por la estación anterior. A partir de la medición el controlador decidirá si el rodamiento debe ser rechazado o no en función del valor medido. Si el objeto no es el adecuado, dispone de un cilindro neumático, como en la FAS-202, para poder rechazar el rodamiento hacia una rampa, de esta manera se extrae el objeto del circuito de ensamblaje y se dispone a introducir un nuevo rodamiento en el sistema. Por el contrario, si el rodamiento es el adecuado, se dispone de un mecanismo de giro junto con una pinza neumática, como en el módulo anterior, para poder hacer el transvase del rodamiento desde la posición de medición hasta posicionarlo en la base, presente en el transfer.

La FAS-204 consta de las siguientes subdivisiones:
medición de la altura y la inserción del rodamiento.



Fig.- 2.16. FAS-204. Fuente: (www.smc.eu).

Medición de la altura

El sistema contempla la posibilidad de alimentar rodamientos con alturas diferentes, por este motivo este módulo cuenta con un cilindro neumático que impulsa el rodamiento hacia un palpador que incorpora un potenciómetro lineal para medir la altura. El módulo anterior (FAS-203) lo posiciona sobre la plataforma de medida que consta de un centrador muy preciso del rodamiento, accionado por un cilindro neumático, esto se necesita para poder realizar de manera adecuada la medición del objeto.

Una vez realizado la medición se puede rechazar la pieza mediante un cilindro neumático que saca la pieza del circuito, como podemos ver en la figura 2.17. o por el contrario, seguir con el proceso de ensamblaje, posicionando el rodamiento dentro de la base.

Se debe de tener en cuenta que el pequeño cilindro centrador esta activado en todo momento, para poder rechazar un rodamiento mediante el cilindro de rechazo, previamente se debe desactivar el cilindro centrador para que el cilindro de rechazo pueda tener vía libre para desplazar hacia el exterior de la plataforma de medición al rodamiento.

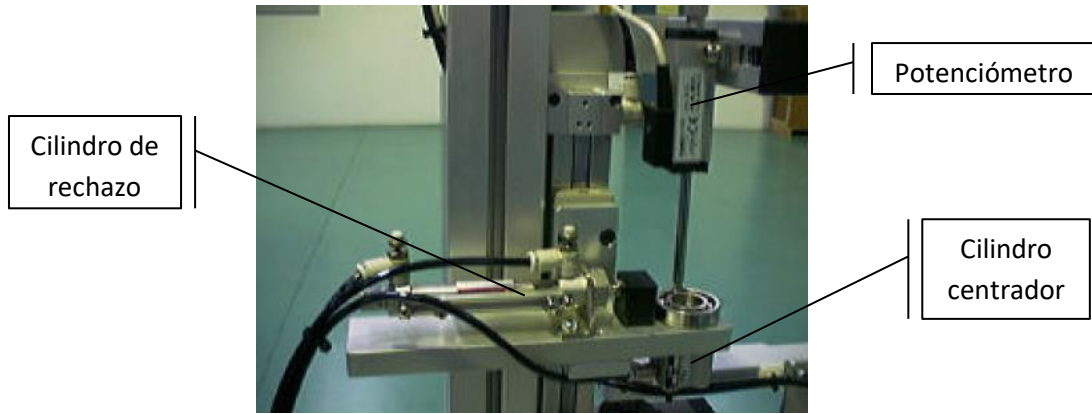


Fig.- 2.17. Medición de la altura del rodamiento. Fuente: (www.smc.eu).

Inserción del rodamiento

Después de que la medición del rodamiento sea correcta, el módulo se encarga de insertar el rodamiento en la base. Este proceso se realiza mediante un manipulador de giro, que realiza un giro de 180º, incorpora un brazo, en la terminación del cual se encuentra una pinza neumática, que sirve de agarre del rodamiento hasta la base. Por otro lado, este manipulador de giro incorpora un cilindro neumático que varía su altura, en el momento de recoger el rodamiento de la posición de medición y cuando procede a la inserción del mismo en la base.

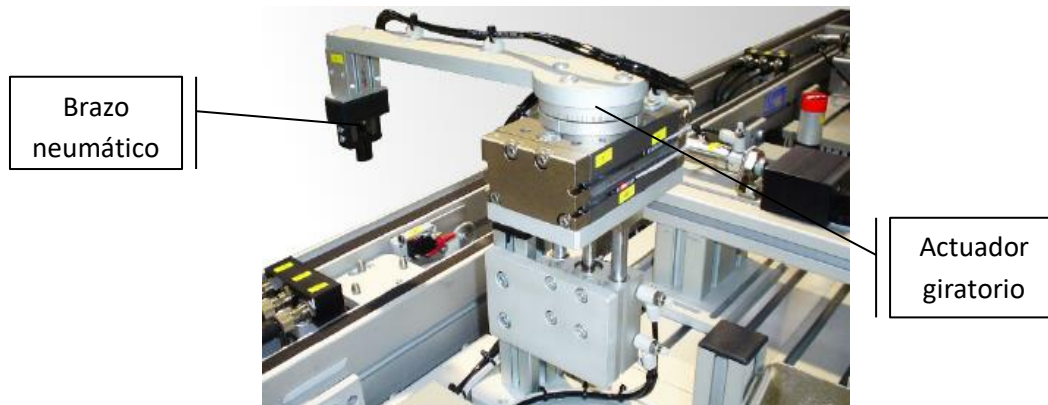


Fig.- 2.18. Inserción del rodamiento en la base. Fuente: (www.smc.eu).

DATOS TÉCNICOS (FAS-204):

Medición de la altura

Actuadores

1 cilindro centrador de simple efecto Ø12, carrera: 5 mm (CB2B12-5S). Controlado por una electroválvula monoestable 3/2.

1 cilindro para el eje vertical sin vástago Ø16, carrera 250 mm (MY1M16G-250), con reguladores de caudal y detectores de posición inicial y final. Controlado por una electroválvula biestable 5/2.

1 cilindro expulsor de piezas de doble efecto Ø10, carrera 40 mm (CD85N10-40), con reguladores de caudal. Controlado por una electroválvula monoestable 5/2.

Sensores

2 detectores magnéticos tipo Reed (D-A93L).

1 potenciómetro lineal GEFAN (PY-2-F-025-S03).

Inserción del rodamiento

Actuadores

1 cilindro compacto de doble efecto con guías Ø16, carrera: 30 mm (MGPM16-30), con reguladores de caudal y detectores de posición inicial y final. Controlado por una electroválvula monoestable 5/2.

1 actuador de giro tipo piñón-doble cremallera Ø20, Ω máx.: 180° (MSQXB20A), con reguladores de caudal y detectores de posición 0° y 180°.

Sensores

2 detectores magnéticos tipo Reed (D-A93L).

2.2.6. FAS-207

Este quinto módulo, FAS-207 (figura 2.19.) continúa con el proceso de montaje de los componentes del mecanismo de giro. Esta vez se añade al sistema los ejes, el módulo aporta más flexibilidad al sistema en cuanto al tipo de productos que podemos obtener, ya que los ejes pueden ser de dos materiales diferentes, aluminio o nylon.

Esta estación es una de las más complejas en cuanto a sus subdivisiones, incorpora una comprobación del tipo de material, orientación y presencia del eje. El control del módulo coordina las operaciones de selección de pieza conforme las órdenes proporcionadas por el controlador maestro responsable de la gestión de la producción.

La confirmación de la presencia, el material y la orientación del eje se lleva a cabo por una serie de sensores (un detector inductivo, un detector capacitivo y un cilindro neumático con detector magnético al final de su carrera, respectivamente).

Las subdivisiones de la FAS-207 son las siguientes: plato divisor, alimentador de ejes, comprobación de la orientación y detección del material.

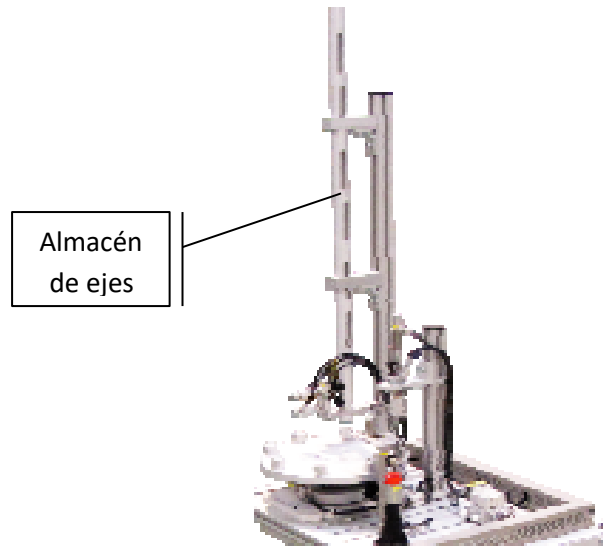


Fig.- 2.19. FAS-207. Fuente: (www.smc.eu).

Plato divisor

El plato divisor consta de ocho posiciones diferentes en las cuales los ejes van pasando, para proceder a su comprobación, hasta ser seleccionados como aptos o no. Este componente sirve como sistema de movimiento giratorio, de forma que cada movimiento de giro corresponde con un giro de unos grados determinados, en este caso 45°.

De las ocho posiciones, tres de ellas están libres, las otras cinco tiene funciones para clasificar los ejes que son introducidos al plato. En la primera posición se incorpora el eje al plato, la segunda posición es libre. Seguidamente la tercera posición se encarga de comprobar la orientación del eje mediante un cilindro neumático que incorpora un sensor magnético tipo *reed* en su final de carrera. En la cuarta posición encontramos dos sensores, un sensor capacitivo que sirve para detecta la presencia de objetos y un sensor inductivo que genera una señal en el momento que detecta un objeto metálico (aluminio) y que esta a cero cuando no lo detecta (nylon o sin presencia). En la quinta posición, en función de las señales generadas por los sensores y las consignas determinadas por el usuario, el eje puede ser expulsado del plato por el módulo siguiente (FAS-208), en caso de que las señales generadas coincidan con las consignas, el eje sigue su camino en el plato. La sexta posición es libre y la séptima es la posición que se encarga de trasladar los ejes correctos a la base que se encuentra en el palet. Este transporte se hace mediante la ayuda de la siguiente estación como en el caso del rechazo de los ejes incorrectos. Por último, la octava posición es libre y se vuelve a efectuar el circuito.

Para conseguir este efecto de giro, el plato dispone de un cilindro empujador con movimiento oscilante. Por otro lado, dispone de dos cilindros de tope que funcionan alternativamente, uno de ellos móvil que sujeta el plato mientras se produce el giro y otro fijo que lo bloquea cuando el movimiento ha finalizado, de esta manera el cilindro empujador puede volver a su posición inicial a la espera de una nueva orden de giro. En la figura 2.20. podemos apreciar el cilindro de avance y el de bloqueo, que funcionan complementariamente.



Fig.- 2.20. Cilindros del plato divisor. Fuente: (www.smc.eu).

Alimentador de ejes

Los ejes son almacenados en un alimentador por gravedad. Como podemos ver en la figura 2.19., estos están quietos hasta el momento de activar la electroválvula que controla, al mismo tiempo, los dos cilindros que sujetan los ejes del alimentador, se pueden apreciar en la figura 2.21. Tienen un comportamiento complementario, en el momento de activarse uno el otro se desactiva. Esto provoca que cuando el inferior se desactiva, cae un eje al plato, el superior se activa y sujeta al resto de los ejes almacenados. Posee un almacenamiento máximo de 17 ejes que, como hemos comentado pueden ser de diferente material.

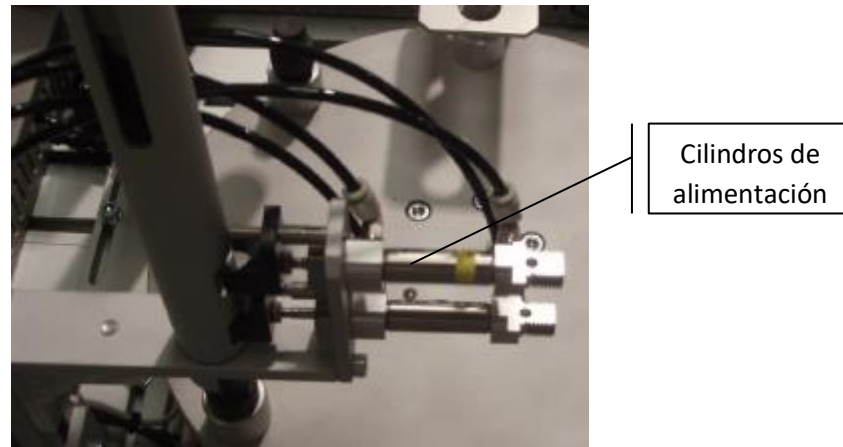


Fig.- 2.21. Cilindros de alimentación. Fuente: (www.smc.eu).

Comprobación de la orientación

El eje a introducir no presenta una forma simétrica, esto obliga al sistema a desarrollar un proceso mediante el cual poder comprobar dicha orientación y si proviene de la posición de alimentación con la orientación correcta. Para ello se utiliza el cilindro neumático mostrado en la figura 2.22., que dispone de un detector magnético que permite discernir si en el avance contacta con el eje o, por el contrario, llega al final de su carrera debido a que el eje ha sido introducido al sistema con la orientación correcta.

Si el eje no posee la orientación correcta el cilindro no detecta la señal de su final de carrera, si esta señal no se activa en un tiempo determinado después de activar el cilindro, el controlador recibe la señal de que el eje se encuentra en una posición errónea.

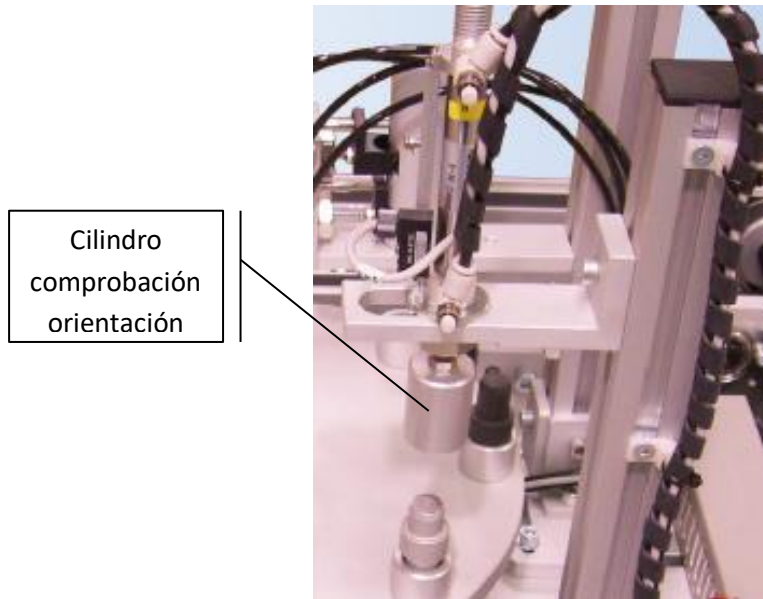


Fig.- 2.22. Cilindro de comprobación de la orientación. Fuente: (www.smc.eu).

Detección del material

Una vez comprobada la orientación del eje, se procede a la detección del material que lo forma para poder discernir entre los ejes que deseamos incorporar a nuestro mecanismo de giro. Para poder distinguir los dos posibles materiales, el sistema cuenta con dos sensores diferentes. Por un lado, podemos observar en la figura 2.23., un sensor capacitivo que crea una señal eléctrica en el momento que hay una presencia, por el otro lado, podemos apreciar un sensor inductivo que genera la señal eléctrica en el momento que detecta un objeto de metal, en este caso aluminio. Con la ayuda de estos dos sensores, podemos distinguir entre si hay presencia de eje, si el eje es de plástico o si por el contrario es de aluminio.

Con esta información podemos proceder a rechazar o incorporar al sistema el eje en proceso, dependiendo de la preferencia que tengamos en el momento.



Fig.- 2.23. Detectores de clasificación del material. Fuente: (www.smc.eu).

DATOS TÉCNICOS (FAS-207)

Plato divisor

Actuadores

1 cilindro compacto de doble efecto Ø25, carrera: 40 mm (CDQ2B25-40D), con reguladores de caudal y detector de posición final. Controlado por una electroválvula monoestable 5/2.

2 cilindros compactos de doble efecto Ø16, carrera: 10 mm (CQ2B16-10D). Controlados por una única electroválvula monoestable 5/2.

Sensores

1 detector magnéticos tipo Reed (D-A93L).

Alimentador de ejes

Actuadores

2 cilindros de doble efecto Ø10, carrera: 10 mm (C85N10-10). Controlados por una única electroválvula monoestable 5/2.

Comprobación de la orientación

Actuadores

1 cilindro de doble efecto Ø12, carrera: 50 mm (CD85N12-50^a), con reguladores de caudal y detector de posición final. Controlado por una electroválvula monoestable 5/2.

Sensores

1 detector magnéticos tipo Reed (D-A93L).

Detección del material

Sensores

1 detector inductivo: OMRON (E2A-M18KS08-WP).

1 detector capacitivo: OMRON (E2K-X4MF1).

2.2.7. FAS-208

En este último módulo (figura 2.24.) se lleva a cabo la selección de los ejes que son suministrados por el módulo anterior (FAS-207). Dependiendo de las señales generadas en las etapas anteriores, el sistema procede a rechazar o a introducir los ejes que se han alimentado, previamente, en el sistema. Teniendo en cuenta el material detectado por los sensores y la elección previa de estos que el usuario ha caracterizado, el sistema se encarga de, por un lado, si la orientación del eje es incorrecta o el material detectado no coincide con el elegido, rechazar el eje. Por otro lado, si la orientación es correcta y el material del eje coincide con el elegido por

el usuario, el módulo FAS-208 incorpora el eje dentro del mecanismo de giro, que se encuentra en el *palet*.

Este módulo también cuenta con varias subdivisiones, entre las que se encuentran las siguientes: inserción del eje en el montaje y rechazo del eje.

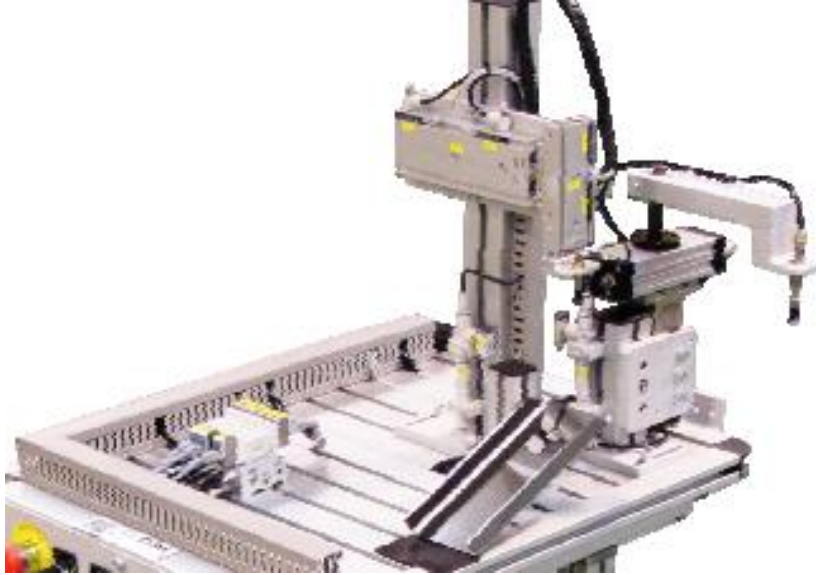


Fig.- 2.24. FAS-208. Fuente: (www.smc.eu).

Inserción del eje en el montaje

La inserción del eje, llevada a cabo desde la última posición del plato divisor del módulo anterior, es desarrollada por un manipulador del tipo roto-lineal. Este cilindro roto-lineal hace posible que con un solo componente se pueda recoger el eje, trasladarlo a la posición de descarga (posición de parada en el transfer lineal) e insertarlo dentro del rodamiento que se encuentra dentro de la base sobre el palet. Este cilindro ofrece la posibilidad de controlar de forma independiente los cuatro movimientos que puede desarrollar, es decir, la salida y entrada del vástago, así como el giro a derechas e izquierdas del mismo.

Este mecanismo presenta en su terminación una ventosa encargada de sujetar el eje en el momento del traslado desde el plato hasta la base emplazada en el palet que se encuentra parado en la cinta delante del módulo. Esta ventosa cuenta con un eyector de vacío, al igual que los otros componentes que poseen ventosas.

También cuenta con un vacuostato que controla el nivel de presión que es creado por el eyector de vacío y lo notifica poniendo a '1' un tag creado en el PLC. En la figura 2.25. podemos apreciar esta subdivisión del módulo.



Fig.- 2.25. Manipulador de inserción de ejes. Fuente: (www.smc.eu).

Rechazo de ejes

Como se ha mencionado anteriormente, este módulo tiene el potencial de poder trabajar en un nivel superior de gestión del sistema automatizado haciendo la elección del material del eje que va a ser montado junto con el resto de componentes para formar el mecanismo de giro final. Esto implica la necesidad de tener algún elemento de rechazo del eje en el caso que éste sea de algún tipo (material) no deseado, operación llevada a cabo en la quinta posición del plato divisor del módulo anterior y que implica un manipulador que recoja el eje del mencionado plato, dependiendo de la señal de mando recibida.

Este elemento tiene forma de manipulador cartesiano de dos ejes, en el extremo final de los mismos se encuentra una ventosa de vacío para sujetar al eje por su parte superior. Cada uno de los ejes está compuesto de un cilindro neumático de vástagos paralelos usados para elevar al eje y trasladarlo hasta la rampa de expulsión. El eje es sujetado mediante un sistema de vacío consistente en una ventosa de succión, igual al observado en los módulos anteriores.

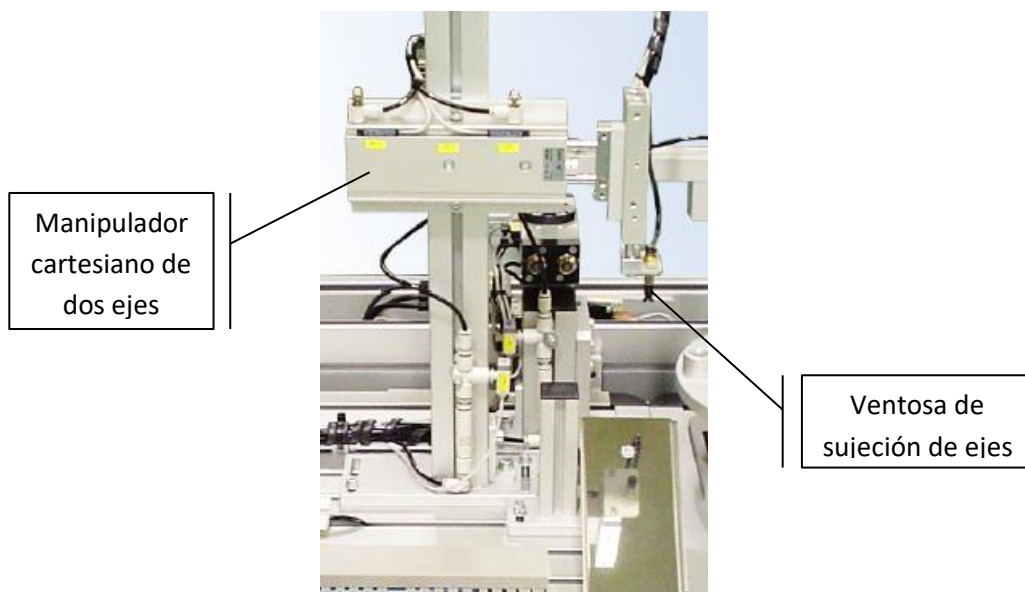


Fig.- 2.26. Elementos de rechazo de ejes. Fuente: (www.smc.eu).

2.3. Requerimientos funcionales

REQUISITO	DESCRIPCI3N
RF01	La botonera se utilizar3 unicamente para entrada en remoto y condiciones iniciales.
RF02	Cada c3lula/m3dulo funcionar3 solo cuando se encuentre en remoto.
RF03	La secuencia de cada pareja de c3lulas funcionar3 cuando las dos se encuentren en autom3tico y el palet este en posici3n de recogida.
RF04	Cuando las c3lulas pasan a remoto, directamente pasan a estar en estado manual.
RF05	El concepto de manual y autom3tico, se aplica a todo el conjunto de la c3lula en cuesti3n y no a cada elemento por separado.
RF06	Si damos la orden de manual y autom3tico al mismo tiempo el sistema se establece en manual.
RF07	El sistema podr3 funcionar en modo manual aun que no est3 el palet en su posici3n normal.
RF08	La transici3n de modo autom3tico a manual siempre se har3 en el momento que el sistema haya acabado su secuencia actual.
RF09	La transici3n de modo manual a autom3tico ser3 de forma directa.
RF10	Se podr3 introducir un nuevo palet para iniciar el ensamblado de una nueva pieza, en el momento que la segunda pareja de c3lulas empiece su secuencia.
RF11	Las diferentes parejas de c3lulas tienen sus programas diferenciados y se controlan por 3 aut3matas diferentes, distribuidos por toda la planta. Pudi3ndose controlar tambi3n desde un solo punto de control, si as3 se desea.
RF12	Todos los programas de las diferentes parejas de c3lulas comparten las rutinas de comportamiento de los elementos que poseen.
RF13	Estas parejas, distribuidas a lo largo de toda la planta, se comunican mediante "tags" de mensaje, para hacer la comunicaci3n y la producci3n de planta m3s eficiente.
RF14	Todas las parejas de c3lulas presentes en el sistema, posee una secuencia de condiciones iniciales, en la cual todos los elementos vuelven a su posici3n inicial.
RF15	Todos los elementos con posibilidad de detectar anomal3as en su funcionamiento, poseen alarmas que nos alertan de posibles fallos en su comportamiento.
RF16	En el momento que salte una alarma de un elemento en concreto, inmediatamente salta la alarma general de dicha c3lula.
RF17	Una vez el sistema pasa a estado de alarma los elementos quedan inutilizados hasta que se establezca la situaci3n normal y los estados de alarma, tanto del elemento, como del sistema sean reseteados.

RF18	Los estados de alarma se reestablecen con el botón de <i>reset</i> respectivo, siempre y cuando la condición de alarma haya desaparecido.
RF19	Las alarmas de detectores de finales de carrera se han parametrizado dependiendo de la velocidad de avance de los respectivos cilindros, siempre dentro de los límites de seguridad.
RF20	La falta de piezas en alguna de las parejas de células es considerada como una alarma y el sistema no podrá seguir con su funcionamiento habitual hasta que este estado este desactivado y el sistema disponga de nuevo material.
RF21	Todas las parejas de células poseen un estado de mantenimiento, en el cual se accede solo desde el modo manual.
RF22	El modo de mantenimiento nos permite poder controlar, en modo manual, cualquier componente, estando este en estado de alarma. Esto nos permite poder revertir el estado de alarma y poder " <i>resetearla</i> " para poder seguir con el proceso.

Tabla 2.1. Contenido de los requerimientos funcionales del sistema. Fuente: (propia).

En la tabla 2.1. podemos observar los requerimientos funcionales del sistema. Estos son de necesario cumplimiento, tanto por parte del sistema, como por parte del usuario, para poder alcanzar una buena práctica (GMP) y conseguir una producción eficiente.

2.4. Requerimientos de diseño

CLASIFICACIÓN	REQUISITO	DESCRIPCIÓN
Ventanas y navegación	RD01	El SCADA tendrá un diseño que se centrará en la comprensión del proceso y la sencillez, buscando una vertiente puramente funcional.
	RD02	El sinóptico principal mostrará una pantalla con opción de poder navegar por los diferentes sinópticos del sistema. Accediendo a través de los botones disponibles.
	RD03	En el sinóptico principal podremos acceder a diferentes ventanas. Pulsando los diferentes botones y accesos directos navegaremos por los sinópticos de las diferentes parejas de células, la pantalla de alarmas, pantalla de información, registros, etc.
	RD04	Las pantallas de las parejas de células tendrán acceso a diferentes ventanas en las que podremos observar el estado de los elementos, manipularlos dependiendo en el modo de funcionamiento que se encuentre las células y observar el estado de la secuencia que este llevando a cabo.

	RD05	Los estados de las secuencias de cada pareja de células las podremos observar mediante el acceso a las ventanas de los procesos en curso, este estado se podrá ver reflejado en un GRAFCET, en el cual podremos ver la ejecución del SFC presente en la programación del PLC.
	RD06	En la ventana de cada pareja de células podremos apreciar el estado en el que se encuentra cada célula que conforma cada pareja. El estado de funcionamiento y si hay alarmas presentes.
	RD07	Habrà una ventana de alarma en la que podremos visualizar las alarmas activas e inactivas.
	RD08	Se podrá acceder a cualquier ventana en menos de tres pasos, para garantizar la agilidad en las operaciones.
	RD09	Se habilitará un botón de salida en forma de cruz a todas las ventanas de tipo <i>pop-up</i> , en la esquina superior derecha.
Código de colores	RD10	Los estados de marcha de los elementos serán de color verde.
	RD11	Los estados de paro o de inactividad de los elementos serán de color gris.
	RD12	Los estados de alarma de los elementos serán de color rojo.
Programación PLC	RD13	La secuencia principal del programa de cada pareja de células, así como la de cada sub proceso se llevará a cabo en SFC para mejorar su comprensión y poder detectar errores de manera más eficiente.
	RD14	El comportamiento de los elementos por separados, en modo manual, las rutinas de <i>"inputs"</i> y <i>"outputs"</i> y por último la rutina de comunicaciones, se llevarán a cabo en lenguaje <i>"ladder"</i> .
	RD15	Para la programación de cada pareja de células, habrá una subrutina programada en <i>"ladder"</i> donde podremos ver que son condiciones simples para avanzar en la programación principal en SFC y cumplir condiciones para continuar con la secuencia.
"Tags"	RD16	El nombre de los <i>"tags"</i> serán los mismo en el programa del PLC como en el SCADA.
	RD17	El nombre de los <i>"tags"</i> se ha creado para facilitar su comprensión, su localización dentro del sistema y saber el componente del que se trata.

Tabla 2.2. Contenido de los requerimientos de diseño del sistema. Fuente: (propia).

En la tabla 2.2. podemos apreciar los requerimientos de diseño, tanto de la aplicación SCADA, como de la programación del PLC. Estos requisitos se deben llevar a cabo y tener en cuenta por

parte del usuario para poder hacer un uso correcto del sistema y para mejorar la comprensi3n de este con m1s facilidad.

2.5. Metodolog1a de desarrollo

En el presente apartado se explica la metodolog1a y los pasos que se deben seguir para poder desarrollar de manera correcta un proyecto de automatizaci3n como el que se esta desarrollando. Desde el an1lisis inicial de la situaci3n del entorno a automatizar, hasta las pruebas finales realizadas en planta.

Cabe destacar que, a parte de la metodolog1a de desarrollo cronol3gica y cl1sica, se ha llevado a cabo una metodolog1a de desarrollo de tipo espiral en la que el tutor hacia a la vez de cliente, donde se le ha ido ense1ando prototipos del producto final peri3dicamente para poder acertar en todo lo que se ha estado implementado en el sistema. Para tener un *feedback* correcto se debe estar constantemente en comunicaci3n con el cliente, en este caso el tutor.

- 1) Agrupaci3n de toda la informaci3n necesaria para entender el comportamiento y funcionamiento general de la planta, as1 como desarrollar un an1lisis previo donde se determine el conjunto de objetivos y el alcance del proyecto.
- 2) Definici3n del fichero de intercambio donde se identifiquen todas las entradas, salidas, estados y ordenes que se puedan extraer del sistema para poder funcionar correctamente.
- 3) Programaci3n de todos los elementos y componentes del sistema por separado para poder conformar el modo de funcionamiento manual.
- 4) Programaci3n de la secuencia/s que ha de seguir el sistema para poder funcionar en el modo autom1tico.
- 5) Creaci3n y montaje de la aplicaci3n SCADA, que permita la visualizaci3n de los estados y la realizaci3n de ordenes del sistema referentes a la programaci3n del PLC.
- 6) Conexi3n del SCADA con el PLC emulado para poder comprobar si las diferentes acciones se van reproduciendo y reflejando de forma correcta.
- 7) Pruebas FAT (Factory Acceptance Testing) para comprobar el correcto funcionamiento del programa. Todo y que no se disponga de una simulaci3n de la planta real, se puede ir variando las entradas y salidas para que parezca una simulaci3n. Esto nos puede permitir simular una situaci3n real y comprobar errores en el funcionamiento para su posterior soluci3n.
- 8) Pruebas SAT (Site Acceptance Testing) llevadas a cabo en el laboratorio, donde se puede comprobar todos los posibles fallos de la programaci3n del PLC como de la aplicaci3n

SCADA y de las comunicaciones entre los diferentes dispositivos tanto “*hardware*” como “*software*”. Una vez identificados dichos errores podemos proceder a la implementaci3n de la soluci3n, para dejar el sistema lo m1s robusto posible, logrando un buen producto final para el cliente.

De la misma manera, en paralelo, se puede ir elaborando la memoria final del proyecto, donde se va a aparecer todos los trabajos realizados y explicados de una manera clara y concisa. En este documento se ver1 reflejado todo el trabajo, por 1ltimo, se har1 una valoraci3n final de todo lo que respecta al proyecto.

2.6. Planificaci3n de las tareas

En el presente apartado se puede observar la programaci3n de los pasos que se han llevado a cabo para la realizaci3n del proyecto. En la tabla 2.3. se aprecia todas las tareas, presentes en el apartado 2.5., estas se han realizado de manera ordenada y cronol3gica, menos en el caso de algunas tareas que se han podido realizar simult1neamente. Este fen3meno se ha producido al final del proyecto, cuando la programaci3n manual y parte de la secuencia autom1tica estaban realizadas. En dicho momento ya se pudo empezar a realizar las pruebas FAT de los componentes, mediante el PLC simulado. Por otro lado, podemos apreciar que una vez acabadas las pruebas FAT, se comenzaron las pruebas SAT, debido a que se tuvo acceso al laboratorio y se pudo volcar la programaci3n sobre la FAS para comprobar su funcionamiento y el programa sobre la maquina real. Con ayuda de esto se pudo ir realizando cambios en la programaci3n hasta depurarla y conseguir su funcionamiento final. Por 1ltimo, como se observa en la tabla la redacci3n de la memoria se ha ido realizando a lo largo de todo el proceso de creaci3n del proyecto.

Actividad	Fecha inicio	Fecha final	Duraci3n (d1as)
Definici3n de objetivos y alcance del proyecto	09-sep	11-sep	3
Funcionamiento y comportamiento general del sistema	12-sep	04-oct	25
Definici3n de todas las se1ales y los <i>tags</i>	07-oct	27-oct	21
Programaci3n del PLC en manual	28-oct	17-nov	21
Programaci3n de la secuencia autom1tica del PLC	18-nov	15-dic	28
Pruebas FAT	21-nov	08-dic	18
Creaci3n de la aplicaci3n SCADA	16-dic	08-ene	24
Pruebas SAT	09-dic	10-ene	33
Redacci3n de la memoria escrita	12-sep	13-ene	124

Tabla 2.3. Tareas a realizadas, duraci3n y fecha de inicio y final de cada parte del proyecto. Fuente: (propia).

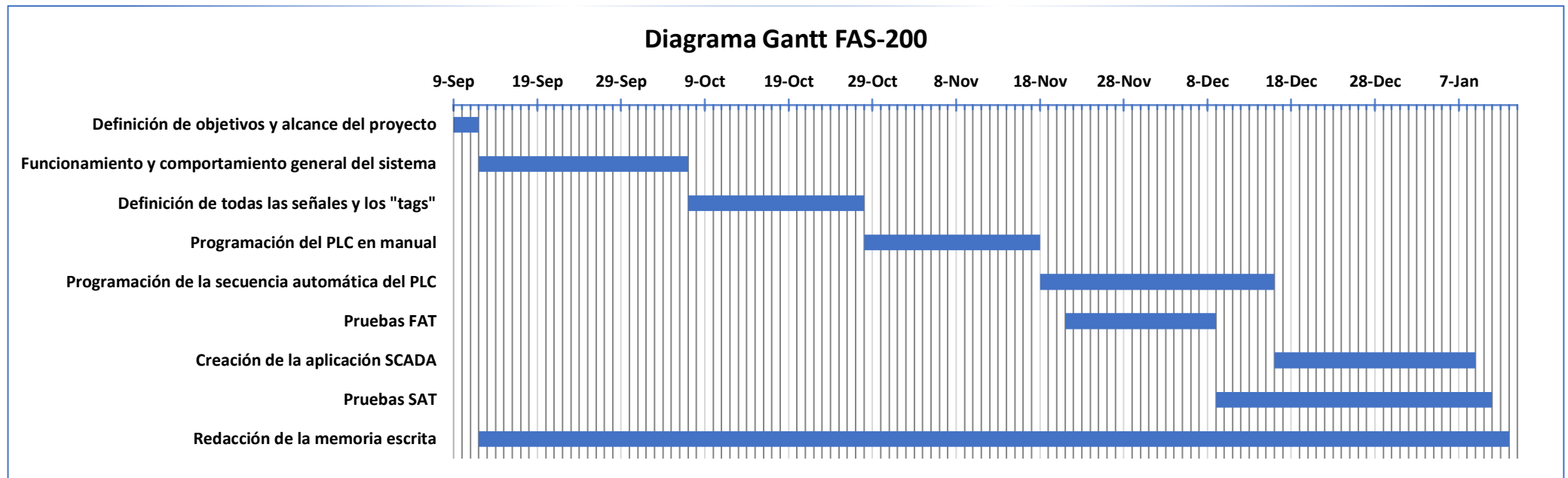


Fig.- 2.27. Diagrama de Gantt del proyecto. Fuente: (propia).

3. Diseño e implementación del proyecto

En el presente apartado se presenta la parte principal del proyecto a automatizar. Empezando por la arquitectura del sistema de control, pasando por la codificación de elemento y finalizando el apartado con la programación del software de control y supervisión.

Proporciona al proyecto de autoridad, veracidad y claridad con relación a los conceptos ya introducidos y nos proporciona todos los datos técnicos y de procedimiento que se han llevado a cabo para lograr el producto final.

3.1. Arquitectura del sistema de control

Cuando hablamos de la arquitectura de un sistema de control podemos estar refiriéndonos a dos conceptos diferentes pero que están ligados el uno al otro. Estamos hablando del “*hardware*” y del “*software*”. A continuación, se desarrolla la arquitectura que poseen ambos dentro del sistema automatizado.

3.1.1. *Hardware* del sistema

Además de las células de automatización del fabricante SMC, comentadas en profundidad en el apartado anterior, de los diferentes “*switch*” de conexión de dispositivos mediante “*Ethernet-IP*”, el sistema también posee los controladores del fabricante Allen-Bradley.

Por un lado, tenemos los equipos presentes en cada célula del sistema y por otro lado el controlador con el que podemos controlar al mismo tiempo más de una célula al mismo tiempo. Los equipos presentes en las células son los MicroLogix 1766-L32BXBA (figura 3.1.) nos sirven de pasarela entre las entradas y salidas físicas del sistema y el controlador principal, esto será comentado más adelante con mayor profundidad.



Fig.- 3.1. MicroLogix 1766-L32BXBA. Fuente: (www.ab.rockwellautomation.com).

El controlador principal que se comunica con los MicroLogix de cada célula para obtener el estado de las entradas y dar ordenes a las salidas para que los actuadores cumplan con su función, son los CompactLogix 1769-L16ER-BB1B (figura 3.2.). Al inicio del proyecto se ha utilizado únicamente un controlador principal para llevar a cabo toda la programación,

posteriormente cuando todo ha funcionado correctamente se ha repartido el peso del programa principal a 3 controlador diferentes, repartidos a lo largo de toda la planta, en nuestro caso el laboratorio.



Fig.- 3.2. CompactLogix 1769-L16ER-BB1B. Fuente: (www.ab.rockwellautomation.com).

3.1.2. Software del sistema

En cuanto a la parte de *software*, esto implica todos los programas informáticos que se han empleado para poder llevar a cabo la programación de los controladores y de la aplicación de supervisión y control del sistema (SCADA).

En primer lugar, cabe destacar que, se ha empleado una máquina virtual con un sistema operativo *Windows 7*, proporcionada por la universidad. Esta máquina virtual posee todos los programas que se han necesitado a lo largo de la creación del proyecto. A continuación, podemos verlos:

- **RSLogix 500 Pro:** se ha utilizado para poder programar los controladores de cada célula, dicho *software* es con el que se programan los autómatas de la serie *MicroLogix*.
- **RSLogix 5000:** este programa se ha utilizado para poder programar el controlador principal del sistema, es el que recoge la información de los registros del *MicroLogix*, lo procesa y lleva a cabo todas las rutinas de programación sobre el sistema.
- **RSLink Classic:** programa que nos permiten definir los *drivers* de comunicación entre los diferentes componentes del sistema. También permite establecer conexiones mediante protocolos DDE/OPC, para poder crear conexión entre el controlador y el software de supervisión y control (SCADA).
- **RSLogix Emulate 5000 Chassis:** herramienta para poder crear bloques de entrada y salida y controladores virtuales, muy útil para las primeras etapas donde no se había probado todavía la programación sobre el controlador real.
- **Wonderware InTouch:** programa para la creación de aplicaciones SCADA, de supervisión, control y adquisición de datos. Como hemos comentado anteriormente, mediante *RSLink* se ha podido establecer comunicación con el controlador y poder controlar el sistema remotamente desde el PC, mediante una interfaz mucho más agradable e intuitiva para el usuario.

3.1.3. Comunicaciones SCADA - Controlador (PLC) - Proceso

En el actual apartado se procede a explicar el método/protocolo de comunicación llevado a cabo para poder conectar todo el *hardware* a la misma red, la configuración de los diferentes *software* anteriormente comentados y la nomenclatura que se ha creado para poder obtener una configuración y asignación de registro eficiente y fácil de manejar.

En la figura 3.3. podemos apreciar un esquema de cómo se encuentra conectado todo el *hardware* del sistema anteriormente comentado.

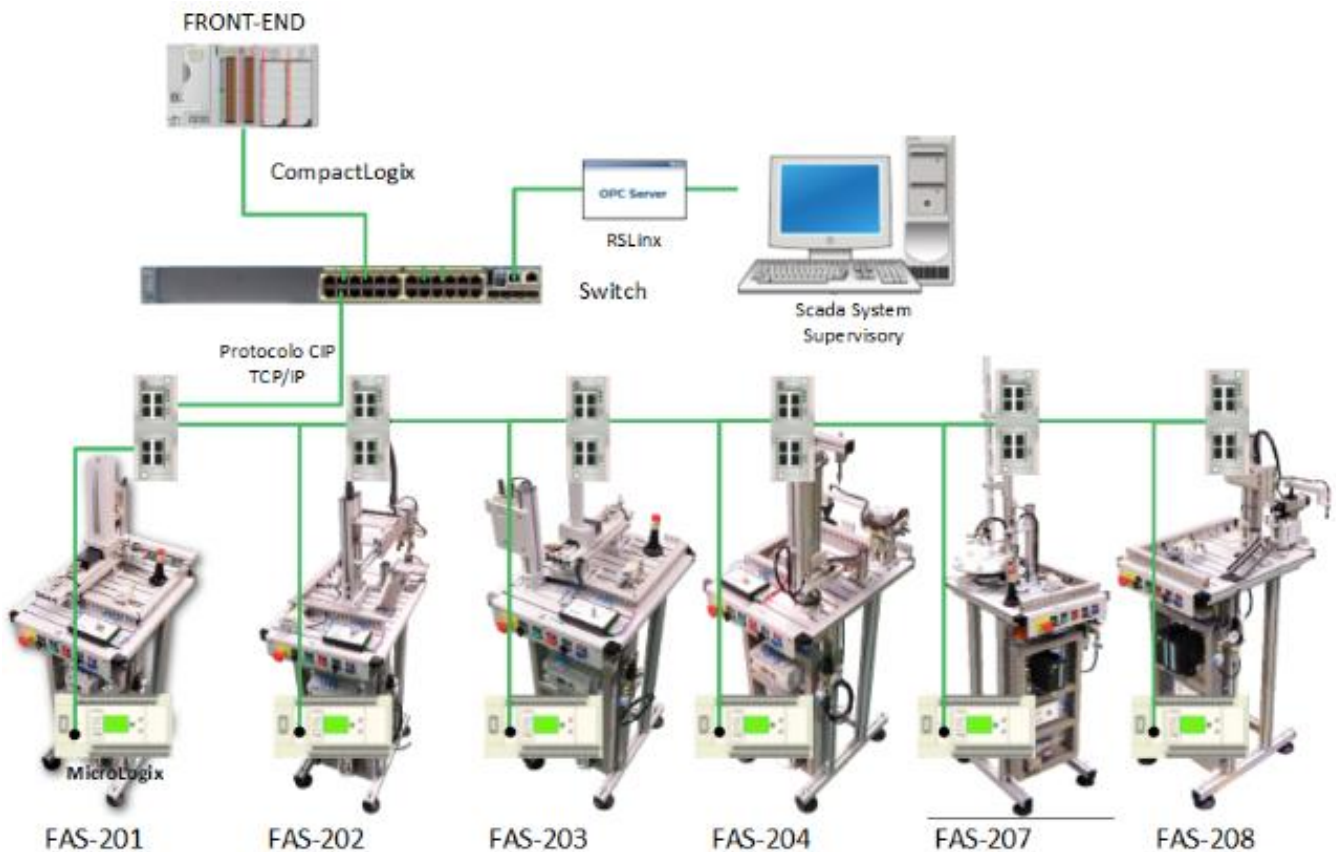


Fig.- 3.3. Esquema general de la conexión de la red del sistema. Fuente: (propia).

Como comprobamos en la figura, todas las comunicaciones se producen mediante *Ethernet/IP*, para hacer efectiva esta comunicación todos los equipos de la red han sido previstos de una dirección IP determinada. Los PLC's han sido dotados de una IP fija para poder facilitar el acceso a ellos en el momento de efectuar pruebas y puestas en marcha. Por otro lado, para poder identificar esta red, la IP de los equipos debe estar en el rango de (130.130.130.XXX).

Los PLC's tiene las siguientes direcciones IP:

- FAS 201 (*MicroLogix*): 130.130.130.1
- FAS 202 (*MicroLogix*): 130.130.130.2
- FAS 203 (*MicroLogix*): 130.130.130.3
- FAS 204 (*MicroLogix*): 130.130.130.4
- FAS 207 (*MicroLogix*): 130.130.130.7
- FAS 208 (*MicroLogix*): 130.130.130.8
- PLC Front-End (*CompactLogix*): 130.130.130.28

Solo nos queda por configurar la red del ordenador desde donde se va a controlar todo el sistema, al haber más de un ordenador en el laboratorio, podemos configurarlos con la IP que deseemos siempre y cuando estén dentro del mismo rango, en la figura 3.4. podemos ver como ha sido configurada la IP del ordenador desde el que se han efectuado las pruebas en el laboratorio.

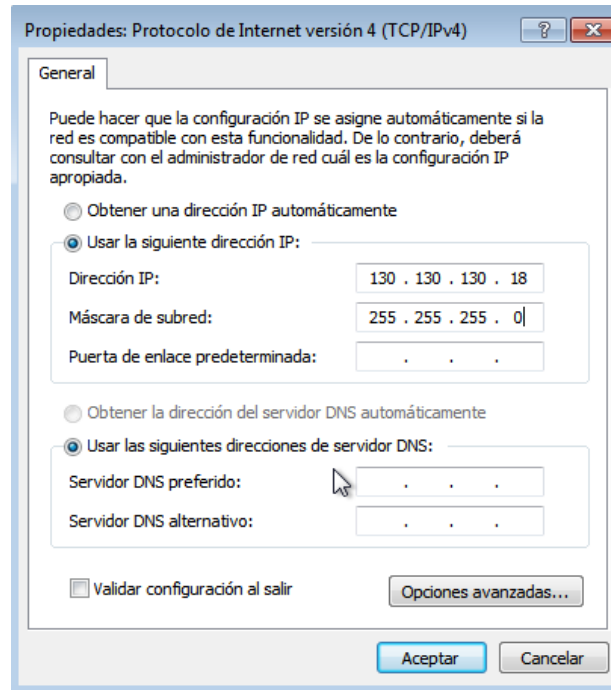


Fig.- 3.4. Configuración IP del PC. Fuente: (propia).

Una vez configuradas las direcciones IP podemos hacer uso de *RSLinx Classic* que nos permitirá poder configurar el driver de *Ethernet IP* para poder identificar a todos los dispositivos de la red que se encuentren en el mismo rango de IP que el PC de donde estamos configurando el driver. En la figura 3.5. podemos ver como se ha configurado el driver y en posteriormente, en la figura 3.6. se puede apreciar como el *software* ya ha identificado los equipos presentes en la red y los muestra en pantalla.

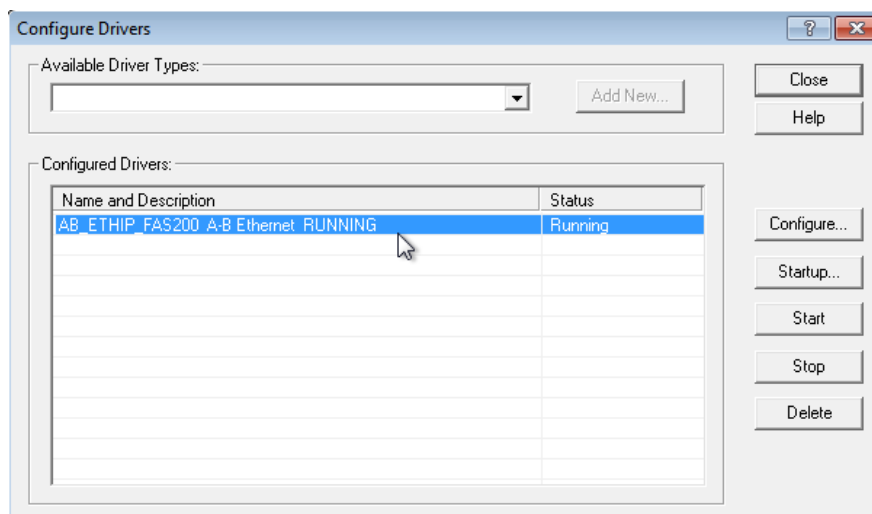


Fig.- 3.5. Driver de comunicación en el entorno de RSLinx Classic. Fuente: (propia).

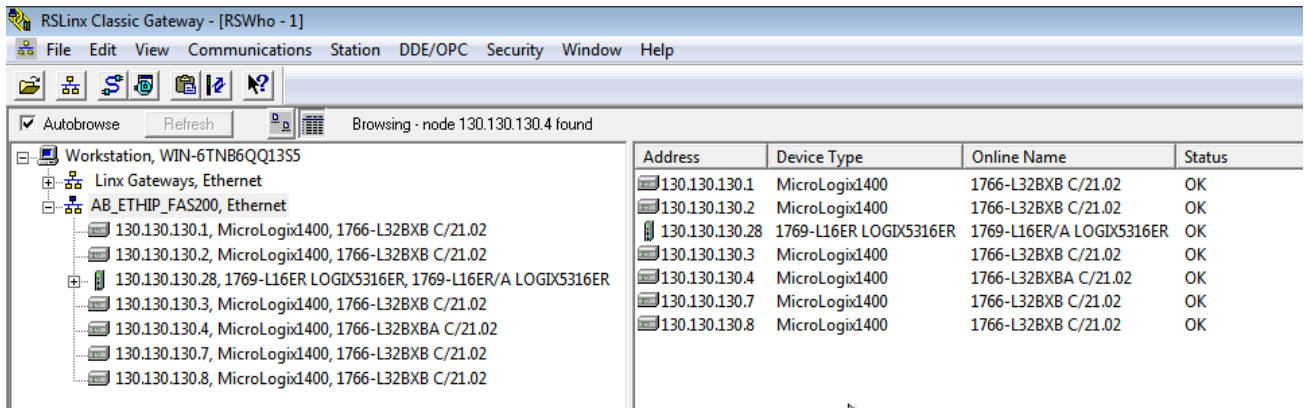


Fig.- 3.6. Dispositivos disponibles dentro de la red configurada mediante el driver Ethernet/IP. Fuente: (propia).

A parte de poder interconectar dispositivo, *RSLogix* también permite asignar *Topics* al controlador que deseemos, esto es utilizado posteriormente para poder realizar conexiones vía DDE/OPC y de esta manera ser capaces de conectar el controlador del sistema con el *software* de supervisión y control (SCADA), más adelante se desarrollará este concepto con mayor profundidad.

Por otro lado, los *PLC's* de las FAS (*MicroLogix*) se deben estar comunicando constantemente con el *PLC Front-End (PLC Maestro)*. Esto es debido a que los estados de las entradas y las salidas físicas del sistema, las recibimos mediante el estado de los registros de entrada y salida de los *MicroLogix* que hay repartidos por todas las FAS. Por este motivo se ha cargado en todos los *PLC's* de las diferentes células un programa *dump* que asigna el estado de las entradas de, por ejemplo, la FAS201 a unos registros que serán leídos en el *PLC Maestro*, lo mismo para las salidas de esta célula y sucesivamente para todas y cada una de las células.

La codificación de los registros que se ha creado ha sido la que se puede observar en la tabla 3.1.

N25	Entradas físicas del sistema
N26	Salidas físicas del sistema
N25.X	X = numero final de la FAS 20X
N25.1	Entradas físicas del sistema de la FAS201.
N26.1	Salidas físicas del sistema de la FAS201.

Tabla 3.1. Codificación de los registros de enlace con los controladores *MicroLogix*. Fuente: (propia).

Los programas creados con el *software RSLogix 500 Pro* los he llamado *dump* ya que simplemente sirven para activar una salida en el momento que un sensor se active y para que una salida se active en el momento que deseemos activar alguno de los actuadores.

En las figuras 3.7. y 3.8. se puede apreciar con más facilidad este concepto.

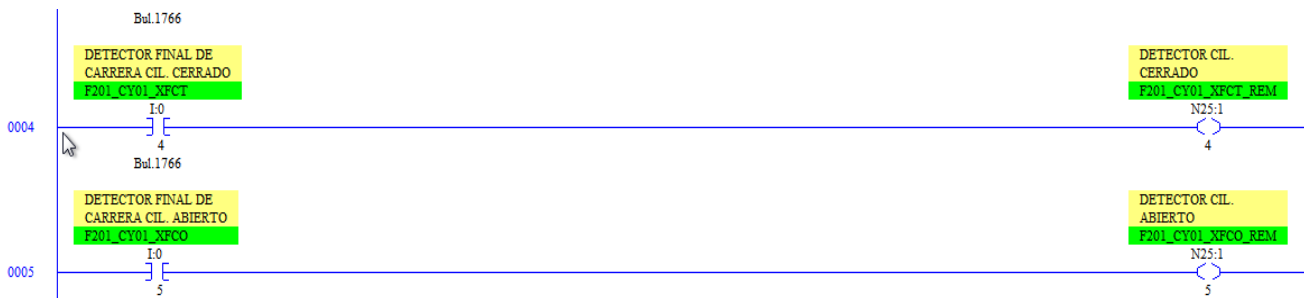


Fig.- 3.7. Asociación de los registros de las entradas físicas del sistema con los registros de enlace para las entradas del controlador principal.

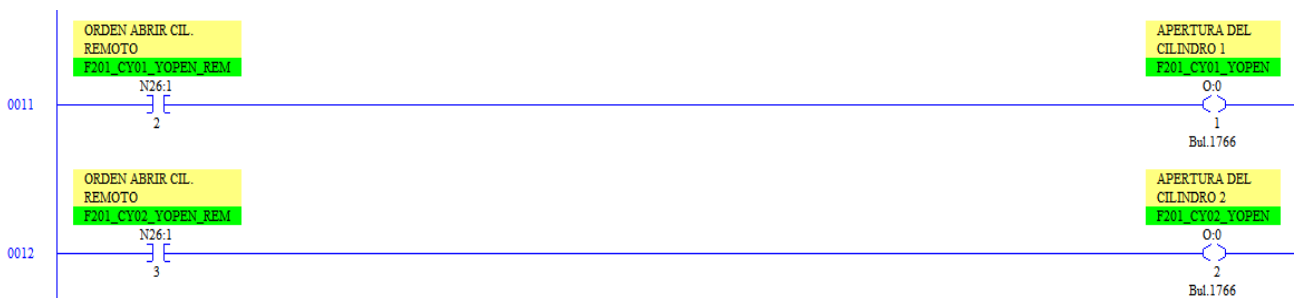


Fig.- 3.8. Asociación de los registros de enlace de las salidas del controlador principal con los registros de las salidas físicas del sistema. Fuente: (propia).

En la figura 3.7. se aprecia como la activación de la entrada física real fuerza la activación de una de las posiciones del registro N25:1, perteneciente, como vemos en la tabla 3.1, a las entradas de la FAS201.

En la figura 3.8. comprobamos que la activación de alguna de las posiciones del registro N26:1, observando la tabla 3.1. vemos que pertenece a las salidas de la FAS201, activa la salida física real, que es finalmente lo que queremos lograr.

En definitiva, el controlador *CompactLogix* solo opera con los registros pertenecientes a la tabla 3.1. esto han servido para hacer de pasarela con las entradas y salidas reales del sistema.

Dichos registros pasarela, están presentes en los programas de los *MicroLogix*, para poder recibir su información en el controlador principal lo logramos mediante la utilización de la función *MESSAGE* presente en el *CompactLogix*.

En la figura 3.9. podemos ver el aspecto de la función de mensajería que se utiliza en el controlador principal para poder obtener información de los estados de los sensores y actuadores.

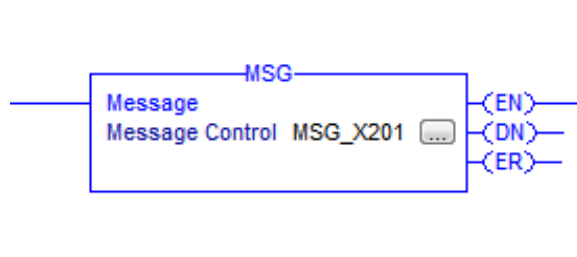


Fig.- 3.9. Función *MESSAGE* del *CompactLogix*. Fuente: (propia).

En la figura 3.10. aparecen gran parte de la mensajería utilizada en el sistema.



Fig.- 3.10. Subrutina de mensajería, comunicaciones entre los MicroLogix y el CompactLogix principal.

Fuente: (propia).

Se ha creado un par de mensajes para cada FAS (entradas y salidas), menos para la FAS204, que posee tres, dos para las entradas y una para la salida, esto es debido a que posee entradas discretas y una entrada analógica.

La figura 3.11. muestra el interior de la instrucción de mensajería y los parámetros que se deben configurar para llevarla a cabo de manera adecuada.

Message Configuration - MSG_X201

Configuration Communication Tag

Message Type: SLC Typed Read

Source Element: N25:1

Number Of Elements: 1

Destination Element: XFAS201

New Tag...

Enable Enable Waiting Start Done Done Length: 0

Error Code: Extended Error Code: Timed Out

Error Path:

Error Text:

Aceptar Cancelar Aplicar Ayuda

Fig.- 3.11. Configuración de la mensajería. Fuente: (propia).

Primero aparece el *Message Type* en el que se elige la opción *SLC Typed Read*, esta opción es la adecuada para poder establecer comunicación entre controladores de la familia *CompactLogix* y *MicroLogix*. En el momento que vemos que el registro es el N25 sabemos que procede de unas entradas, es por esta razón que es *Read* ya que estamos leyendo el estado de las entradas físicas del sistema, en este caso de la FAS201.

En el *Source Element* escogemos el registro del que queremos recibir información y en el *Destination Element* escogemos el elemento o *tag* en el que queremos almacenar dicha información recolectada. En este caso, la información la emplazamos en un *tag* de tipo INT que posee 16 posiciones justo como los registros de la FAS.

De esta manera se procede a enlazar la información recogida en el *MicroLogix* con el registro presente en el controlador principal.

La pantalla de comunicaciones que podemos ver en la figura 3.12. es la misma tanto para la lectura como para la escritura. Vemos que debemos señalarle la ruta que debe seguir para poder comunicar con el elemento que le va a aportar información y sobre el que va a poder enviarle información de vuelta.

En el caso de la imagen, esta comunicando con la FAS201.

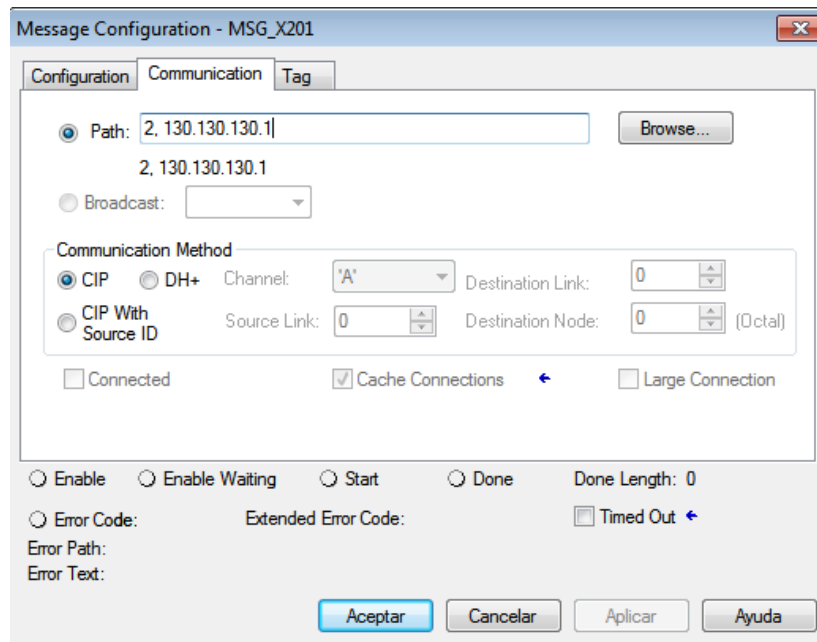


Fig.- 3.12. Configuración de las comunicaciones de la función de mensajería. Fuente: (propia).

3.2. Descomposición del proyecto a automatizar

En el actual apartado se exponen los pasos a seguir y los conceptos necesarios que se ir llevando a cabo para poder realizar un proyecto de manera eficiente y de manera ordenada.

Después de estudiar el sistema de comunicaciones y la arquitectura del sistema, se procede a crear una codificación de los componentes del sistema, de una manera intuitiva y clara para los usuarios y/o personas ajenas al sistema.

Como en el sistema hay componente que poseen las misma características y señales eléctricas se procederá a la creación de UDT's (*User Defined Type*). Esta opción nos ayuda a crear los elementos y sus señales de manera mucho más eficiente y rápida, las señales de un tipo de elemento solo se tendrán que incorporar una vez en el sistema. Una vez se introduzca un elemento de un tipo que ya ha sido introducido solo se tendrá que indicar y ese elemento nuevo poseerá las mismas señales.

Una vez se tienen todas las señales se procede a la programación del controlador. Se ha establecido una programación por subrutinas con una estructura bien clara, estas subrutinas engloban una agrupación independiente de código que realiza acciones diferentes a las demás subrutinas.

Se debe tener en cuenta que existen dos tipos de funcionamiento, el manual y el automático, en el modo manual lo que manda es la simple manipulación de los elementos por separado. En el modo automático el sistema entra en una secuencia que hasta que no sale de ella no se debe tener ningún tipo de interferencia, con la gran excepción de la aparición de las alarmas.

Una vez obtenido toda la programación y después de las pruebas FAT pertinentes se puede proceder a descargar en el controlador la programación llevada a cabo, comprobar el funcionamiento y resolver todas las anomalías que van surgiendo.

3.3. Codificación de los elementos y sistemas

En el presente apartado se procede a explicar la codificación de los elementos del sistema, parte muy importante para poder entender de forma adecuada la programación y los elementos que el sistema posee.

La codificación se ha llevado a cabo con los mismos criterios con los que se han impartido en la asignatura ISA (Integración de Sistemas Automáticos).

Con el objetivo de poder crear una codificación clara, al nombre primero se le añade un "F", seguido del número de la FAS en cuestión, un guion bajo para poder separar la parte del sistema al que pertenecen de la parte donde se establece el componente al que hace referencia. De esta manera todos los componentes del que podamos ver su nombre nos aportarán información al instante, sistema al que pertenece y componente.

Se ha creado una codificación para poder asignar un nombre a cada elemento del sistema dependiendo del tipo que sea, en la tabla 3.2. se puede observar dicha codificación.

Posteriormente, se han creado las UDT's, es por este motivo por el que un componente puede ser de dos o más UDT's diferentes. Por ejemplo, un actuador rotativo (RA) puede ser RA2D o RA3D, dependiendo del componente en cuestión. En la tabla 3.3. podemos apreciarlo.

Componente	Descripción	Componente	Descripción
CY	Cilindros	RA	Actuador rotativo
IS	Sensor inductivo	LS	Final de carrera
VS	Ventosas	GP	Pinzas neumáticas
AS	Detector analógico	CS	Sensor capacitivo

Tabla 3.2. Codificación de los componentes del sistema. Fuente: (propia).

UDT	Descripción
CYS0	Cilindro de simple efecto sin detectores
CYS1	Cilindro de simple efecto con un detector
CYS2	Cilindro de simple efecto con dos detectores
CYD2	Cilindro de doble efecto con dos detectores
DET	Sensor
SC	Ventosas
RA2D	Actuador rotativo con dos detectores
RA3D	Actuador rotativo con tres detectores
GRAP	Pinza neumática
ADET	Sensor analógico
SYS	Sistema

Tabla 3.3. UDT's del sistema. Fuente: (propia).

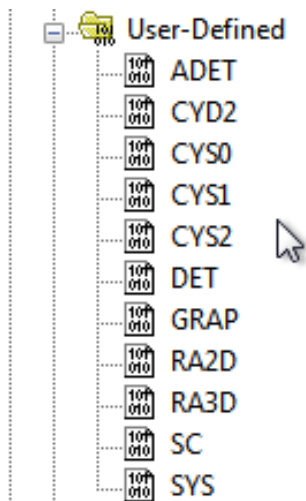


Fig.- 3.13. UDT's presentes en el entorno de programación. Fuente: (propia).

A continuación, en la tabla 3.4. podemos visualizar los componentes del sistema, con su codificación, como se ha comentado anteriormente.

Elemento	Descripción
F201_CY01	Cilindro alimentación de bases
F201_CY02	Cilindro verificación de orientación de bases
F201_CY03	Cilindro de transvase de bases
F201_IS01	Detector inductivo de bases
F202_CY01	Cilindro horizontal de transporte de bases
F202_CY02	Cilindro vertical de transporte de bases
F202_CY03	Cilindro de expulsión de las bases
F202_VS01	Ventosas de transporte de bases
F203_RA01	Act. rot. del brazo transportador de rodamientos
F203_CY01	Cilindro de alimentación de rodamientos

F203_LS01	Final de carrera para detecci3n de rodamientos
F203_GP01	Pinza de transvase de rodamientos
F204_RA01	Act. rot. del brazo transportador de rodamientos
F204_CY01	Cilindro de medici3n de rodamientos
F204_CY02	Cilindro de rechazo de rodamientos
F204_CY03	Cilindro centrador de rodamientos para medici3n
F204_CY04	Cilindro vertical del act. rotativo
F204_AS01	Sensor anal3gico medidor de la altura del rodamiento
F204_GP01	Pinza de transvase de rodamientos
F207_CY01	Cilindros horizontales de alimentaci3n de ejes
F207_CY02	Cilindro de comprobaci3n de la orientaci3n
F207_CY03	Cilindro empujador de avance del plato giratorio
F207_CY04	Cilindro de bloqueo del plato giratorio
F207_IS01	Detector inductivo detector de ejes
F207_CS01	Detector capacitivo detector de ejes
F208_RA01	Act. rot. del brazo transportador de ejes
F208_CY01	Cilindro vertical del act. rotativo
F208_CY02	Cilindro horizontal de rechazo de ejes
F208_CY03	Cilindro vertical de rechazo de ejes
F208_VS01	Ventosa de inserci3n de eje en la base
F208_VS02	Ventosa de rechazo de ejes
FBELT_CY12	Cilindro del <i>stopper</i> de la primera pareja
FBELT_CY34	Cilindro del <i>stopper</i> de la segunda pareja
FBELT_CY34UP	Cilindro elevador de la base
FBELT_CY78	Cilindro del <i>stopper</i> de la tercera pareja
FBELT_LS12	Final de carrera de la cinta para detectar palet
FBELT_LS34	Final de carrera de la cinta para detectar palet
FBELT_LS78	Final de carrera de la cinta para detectar palet

Tabla 3.4. Componentes del sistema. Fuente: (propia).

Una vez tenemos claro la codificaci3n de los componentes y las UDT's a las que pertenecen, se procede a definir la nomenclatura de las se~ales de los componentes, en el apartado 3.5. se puede observar el fichero de intercambio entre el controlador y el SCADA, donde se observan todas las se~ales de cada una de las UDT's definidas anteriormente.

Existen 4 tipos de se~ales, dependiendo de su procedencia y su destino final. La figura 3.14. nos muestra los diferentes tipos de se~ales.

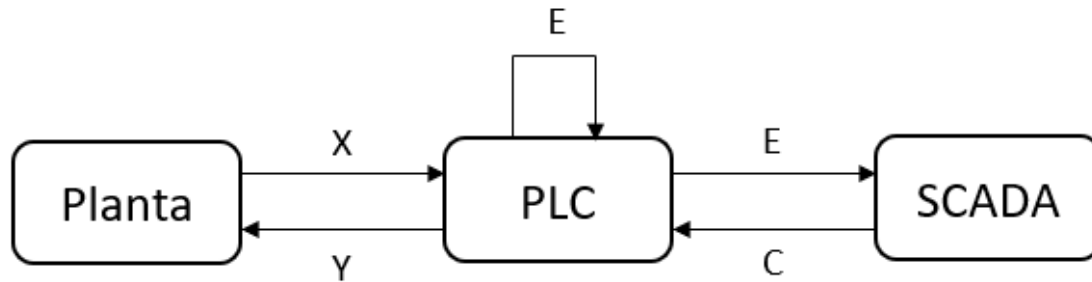


Fig.- 3.14. Tipos de señales del sistema. Fuente: (propia).

Como podemos observar los identificadores de las señales son los siguiente:

- Identificador X: pertenecen a las señales que proceden de planta. Representan a los sensores y detectores.
Ejemplo: señal F201_CY01.XFCO, perteneciente al detector de final de carrera del cilindro 1 de la FAS 201.
- Identificador Y: son las señales que el controlador manda a planta, el controlador las activa según su programación para poder activar los diferentes actuadores.
Ejemplo: señal F201_CY01.YOPEN, perteneciente a la orden de abrir cilindro 1 de la FAS 201.
- Identificador E: estas señales se refieren a los estados internos del controlador y de todos los componentes a los que este está asociado. Estas señales también se mandan al sistema de supervisión y control (SCADA) para que este sistema adquiera el estado de sensores, estados de sistema y situación de los actuadores.
Ejemplo: señal F201.E_MAN, perteneciente al estado de manual del sistema FAS 201.
- Identificador C: por último, estas señales son las que van asociadas a las ordenes que proceden del SCADA, mediante las cuales podemos cambiar de estado actuadores, cambiar el estado y activar condiciones del sistema.

Ejemplo: señal F201_CY01.C_OPEN, perteneciente a la orden de abrir el cilindro 1 de la FAS 201.

Cabe destacar que este tipo de señales poseen dos versiones diferenciadas, dependiendo del modo en el cual estemos dando la orden, manual o automático, esta señal tendrá dos versiones.

Empleando el ejemplo anterior, F201_CY01.C_OPEN, vemos que es la señal de orden de activación del cilindro, si se desea activar dicho cilindro en estado automático, en el momento que se este ejecutando la secuencia en automático, se debe de activar la señal F201_CY01.C_OPEN_A. La incorporación de “_A” al final del *tag* hace que este asociado a la orden en el estado automático.

3.4. Definici3n de la interfaz del sistema de control

Con el desarrollo de los apartados y de la informaci3n aportada se ha podido comprobar que el sistema general a automatizar, FAS200, ya posee un sistema de control individual. Cada c3lula posee su controlador particular, como se sabe, el objetivo de este proyecto es poder crear un sistema conjunto en el cual seamos capaces de poder controlar todas las c3lulas desde un mismo controlador. Debido a esto es necesario poder definir cuando el sistema pasa de funcionar en local, con sus controladores particulares, a poder ser controlador por el *PLC-Front End* principal.

La diferentes FAS poseen una botonera, mediante esta ser3 como podremos pasar el sistema a remoto y poder tener control sobre 3l. Para poder controlarlo en remoto debemos presionar al mismo tiempo el bot3n de *START* y *STOP* al mismo tiempo hasta que se encienda el *LED* azul del bot3n de *RESET* de la botonera. En la figura 3.14. podemos ver el sistema en este estado.



Fig.- 3.15. Botonera de las FAS. Fuente: (www.smc.eu)

El selector de la botonera debe permanecer en estado “auto” si se pasa a estado “man” el sistema sale del estado de remoto.

Por otro lado, se debe tener en cuenta que los controladores particulares de cada FAS siguen en funcionamiento ya que son los que nos permiten poder escribir y leer los estados de sus registros de entradas y salidas, estos controladores nos sirven como una tarjeta de entradas y salida del sistema, al cual esta conectado el controlador principal.

La botonera del sistema, como hemos comentado, sirve para pasar el sistema en remoto, pero tambi3n posee una segunda funci3n. Podremos pasar al estado de condiciones iniciales mediante la manipulaci3n de la botonera. Pulsando al mismo tiempo, siempre y cuando nos encontremos en estado manual, los botones de *START* y *RESET* se entrar3 en el estado de condiciones iniciales que har3 que los actuadores vuelvan a su posici3n inicial.

La 3ltima funci3n que tiene la botonera es la para de emergencia que se puede ejecutar mediante la pulsaci3n de la “seta” de emergencia. Esto provoca el corte de alimentaci3n del controlador y de toda la c3lula, dej3ndola inoperativa.

3.5. Fichero de intercambio entre el controlador y SCADA

Las señales asociadas a la UDT CYS0 son las siguientes:

CYS0			PLC			SCADA		
TAG	TIPO	DESCRIPCIÓN	E	E/S	S	E	E/S	S
C_OPEN	BOOL	Orden de abrir	1					1
C_OPEN_A	BOOL	Orden de abrir automático	1					1
E_OPEN	BOOL	Estado abierto			1	1		
YOPEN	BOOL	Señal de apertura			1			

Tabla 3.5. UDT CYS0. Fuente: (propia).

Las señales asociadas a la UDT CYS1 son las siguiente:

CYS1			PLC			SCADA		
TAG	TIPO	DESCRIPCIÓN	E	E/S	S	E	E/S	S
C_OPEN	BOOL	Orden de apertura	1					1
C_OPEN_A	BOOL	Orden de apertura automático	1					1
C_RESET_ALARM	BOOL	Orden reseteo alarma						
E_ALARM	BOOL	Estado de alarma			1	1		
E_FCO	BOOL	Estado de final de carrera abierto			1	1		
E_OPEN	BOOL	Estado abierto			1	1		
XFCO	BOOL	Señal de entrada final de carrera abierto	1					
YOPEN	BOOL	Señal de salida apertura			1			
TIMER_FCO_CYS1	TIMER	Timer para la activación de FCO						

Tabla 3.6. UDT CYS1. Fuente (propia).

Las señales asociadas a la UDT CYS2 son las siguientes:

CYS2			PLC			SCADA		
TAG	TIPO	DESCRIPCIÓN	E	E/S	S	E	E/S	S
C_OPEN	BOOL	Orden de apertura	1					1
C_OPEN_A	BOOL	Orden de apertura automático	1					1
C_RESET_ALARM	BOOL	Orden de reseteo de alarma	1					1
E_ALARM	BOOL	Estado de alarma			1	1		
E_FCO	BOOL	Estado de final de carrera abierto			1	1		
E_FCT	BOOL	Estado de final de carrera cerrado			1	1		
E_OPEN	BOOL	Estado abierto			1	1		
XFCO	BOOL	Señal de entrada final de carrera abierto	1					
XFCT	BOOL	Señal de entrada final de carrera cerrado	1					

YOPEN	BOOL	Señal de salida apertura			1			
TIMER_FCO_CYS2	TIMER	Temporizador para activación FCO.						
TIMER_FCT_CYS2	TIMER	Temporizador para activación FCT.						

Tabla 3.7. UDT CYS2. Fuente: (propia).

Las señales asociadas a la UDT CYD2 son las siguientes:

CYD2			PLC			SCADA		
TAG	TIPO	DESCRIPCIÓN	E	E/S	S	E	E/S	S
C_OPEN	BOOL	Orden de apertura	1					1
C_CLOSE	BOOL	Orden de cierre	1					1
E_ALARM	BOOL	Estado de alarma			1	1		
E_FCO	BOOL	Estado de final de carrera abierto			1	1		
E_FCT	BOOL	Estado final de carrera cerrado			1	1		
E_OPEN	BOOL	Estado abierto			1	1		
E_CLOSE	BOOL	Estado cerrado			1	1		
XFCO	BOOL	Señal de entrada final de carrera abierto	1					
XFCT	BOOL	Señal de entrada final de carrera cerrado	1					
YOPEN	BOOL	Señal de salida apertura			1			
YCLOSE	BOOL	Señal de salida cierre			1			

Tabla 3.8. UDT CYD2. Fuente: (propia).

Las señales asociadas a la UDT SC son las siguientes:

SC			PLC			SCADA		
TAG	TIPO	DESCRIPCIÓN	E	E/S	S	E	E/S	S
C_TAKE	BOOL	Orden de coger objeto	1					1
C_TAKE_A	BOOL	Orden de coger objeto automática	1					1
C_LEAVE	BOOL	Orden de dejar objeto	1					1
C_LEAVE_A	BOOL	Orden de dejar objeto automática	1					1
C_RESET_ALARM	BOOL	Orden de reseteo de la alarma	1					1
E_TAKE	BOOL	Estado de coger			1	1		
E_ALARM	BOOL	Estado de alarma			1	1		
E_LEAVE	BOOL	Estado de dejar			1	1		
E_OK	BOOL	Estado presión OK			1	1		
XOK	BOOL	Señal de entrada presión OK	1					
YTAKE	BOOL	Señal de salida de coger objeto			1			
YLEAVE	BOOL	Señal de salida de dejar objeto			1			
TIMER_PRES_OK	BOOL	Timer detección presión OK						

Tabla 3.9. UDT SC. Fuente: (propia).

Las señales asociadas a la UDT RA2D son las siguientes:

RA2D			PLC			SCADA		
TAG	TIPO	DESCRIPCIÓN	E	E/S	S	E	E/S	S
C_AHEAD	BOOL	Orden adelante	1					1
C_AHEAD_A	BOOL	Orden adelante automático	1					1
C_RESET_ALARM	BOOL	Orden de reseteo de alarma	1					1
E_AHEAD	BOOL	Estado adelante			1	1		
E_PEND	BOOL	Estado de posición final			1	1		
E_PINI	BOOL	Estado de posición inicial			1	1		
E_ALARM_PINI	BOOL	Estado de alarma por posición inicial.			1	1		
E_ALARM_PEND	BOOL	Estado de alarma por posición final.			1	1		
XPEND	BOOL	Señal de entrada de posición final	1					
XPINI	BOOL	Señal de entrada de posición inicial	1					
YAHEAD	BOOL	Señal de salida adelante			1			
TIMER_PINI_RA2D	TIMER	Timer activación posición inicial.						
TIMER_PEND_RA2D	TIMER	Timer activación posición final.						

Tabla 3.10. UDT RA2D. Fuente: (propia).

Las señales asociadas a la UDT RA3D son las siguientes:

RA3D			PLC			SCADA		
TAG	TIPO	DESCRIPCIÓN	E	E/S	S	E	E/S	S
C_AHEAD	BOOL	Orden adelante	1					1
C_BEHIND	BOOL	Orden atrás	1					1
C_AHEAD_A	BOOL	Orden adelante automática	1					1
C_BEHIND_A	BOOL	Orden de atrás automática	1					1
C_INI	BOOL	Orden de posición inicial	1					1
C_RESET_ALARM	BOOL	Orden reseteo alarma	1					1
E_ALARM_PINI	BOOL	Estado de alarma posición inicial			1	1		
E_ALARM_PEND	BOOL	Estado de alarma posición final			1	1		
E_AHEAD	BOOL	Estada adelante			1	1		
E_BEHIND	BOOL	Estado atrás			1	1		
E_MIDDLE	BOOL	Estado de posición media			1	1		
E_PEND	BOOL	Estado de posición final			1	1		
E_PINI	BOOL	Estado de posición inicial			1	1		
XPEND	BOOL	Señal de entrada de posición final	1					
XPINI	BOOL	Señal de entrada de posición inicial	1					
XPMIDDLE	BOOL	Señal de entrada de posición media	1					
YAHEAD	BOOL	Señal de salida adelante			1			
YBEHIND	BOOL	Señal de salida atrás			1			

Tabla 3.11. UDT RA3D. Fuente: (propia).

Las señales asociadas a la UDT GRAP son las siguientes:

GRAP			PLC			SCADA		
TAG	TIPO	DESCRIPCIÓN	E	E/S	S	E	E/S	S
C_OPEN	BOOL	Orden de abrir	1					1
C_CLOSE	BOOL	Orden de cerrar	1					1
C_OPEN_A	BOOL	Orden de abrir automática	1					1
C_CLOSE_A	BOOL	Orden de cerrar automática	1					1
E_OPEN	BOOL	Estado de apertura			1	1		
E_CLOSE	BOOL	Estado de cierre			1	1		
YOPEN	BOOL	Señal de salida de apertura			1			
YCLOSE	BOOL	Señal de salida de cierre			1			

Tabla 3.12. UDT GRAP. Fuente: (propia).

Las señales asociadas a la UDT DET son las siguientes:

DET			PLC			SCADA		
TAG	TIPO	DESCRIPCIÓN	E	E/S	S	E	E/S	S
E_DET	BOOL	Estado de detección			1	1		
XDET	BOOL	Señal de entrada de detección	1					

Tabla 3.13. UDT DET. Fuente: (propia).

Las señales asociadas a la UDT ADET son las siguientes:

ADET			PLC			SCADA		
TAG	TIPO	DESCRIPCIÓN	E	E/S	S	E	E/S	S
MINS	REAL	Medida			1	1		
XMINS	REAL	Señal de entrada de la medida	1					

Tabla 3.14. UDT ADET. Fuente: (propia).

Las señales asociadas a la UDT SYS son las siguientes:

SYS			PLC			SCADA		
TAG	TIPO	DESCRIPCIÓN	E	E/S	S	E	E/S	S
C_MAN	BOOL	Orden de manual	1					1
C_AUTO	BOOL	Orden de automático	1					1
C_MAINTENANCE	BOOL	Orden de mantenimiento	1					1
C_RESET_ALARM	BOOL	Orden de reseteo de alarma	1					1
C_PIECE_OK	BOOL	Orden de pieza correcta	1					1

E_PIECE_READY	BOOL	Estado de pieza lista para ser alimentada			1	1		
E_ALARM	BOOL	Estado de alarma			1	1		
E_MAN	BOOL	Estado manual			1	1		
E_AUTO	BOOL	Estado automático			1	1		
E_PIECE	BOOL	Estado de falta de pieza			1	1		
E_REMOTO	BOOL	Estado de remoto del sistema			1	1		
E_MAINTENANCE	BOOL	Estado de mantenimiento			1	1		
E_INI_COND	BOOL	Estado de condiciones iniciales			1	1		
XSTART	BOOL	Señal de entrada botón START	1					
XSTOP	BOOL	Señal de entrada botón STOP	1					
XSELECT	BOOL	Señal de entrada selector manual/automático	1					
XRESET	BOOL	Señal de entrada de botón RESET	1					
XREMOTO	BOOL	Señal de entrada de remoto	1					
YLED_REMOTO	BOOL	Led señal estado remoto			1			
YLED_PIECE	BOOL	Led señal falta de piezas			1			

Tabla 3.15. UDT SYS. Fuente: (propia).

3.6. Programa del controlador

Después de estudiar el sistema, su arquitectura, el planteamiento y codificación de todas sus entradas, salidas y señales que intervienen en el sistema, es hora de volcar toda esta información sobre el controlador. En el apartado actual se trata con profundidad toda la programación del controlador, desde la estructura con la que se ha diseñado el sistema, pasando por el tipo de datos empleados y por último profundizando en la lógica que se lleva a cabo, tanto de los componentes en modo manual, como las secuencias del modo automático.

3.6.1. Estructura del controlador

La estructura interna del controlador, programado con el *software RSLogix5000*, esta separada en cuatro programas diferenciados, como podemos ver en la figura 3.16.

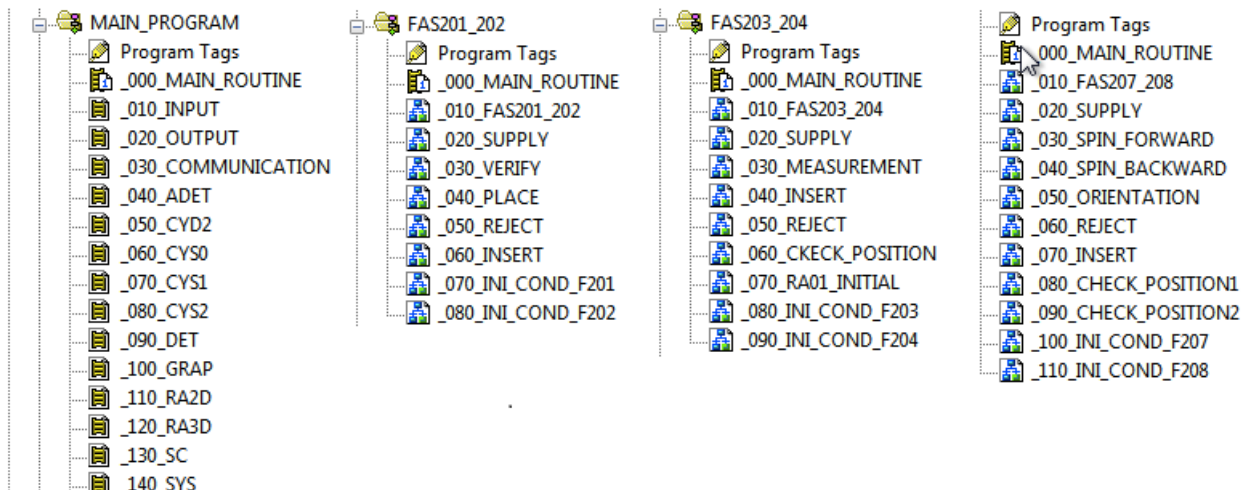


Fig.- 3.16. Estructura de los programas del controlador del sistema. Fuente: (propia).

Esta separación de los programas se ha realizado para tener una mejor perspectiva de los programas y sus respectivas rutinas.

En el programa llamado MAIN_PROGRAM se han programado las rutinas de todas la UDT's del sistema, junto con la rutina de entradas, salidas y las comunicaciones del sistema con los controladores de las células (*MicroLogix*).

La rutina _000_MAIN_ROUTINE es la rutina que se ejecutan a cada ciclo de programa y va llamando al resto para que se vayan ejecutando ordenadamente.

En la rutina _010_INPUT se definen las entradas al sistema provenientes de los registros de los controladores de las células. Por otro lado, en la rutina _020_OUTPUT se definen las salidas del sistema para poder enviar dicha información a los registros de las salidas de los controladores de las células.

Toda la información que se maneja en las dos rutinas de entradas y salidas es posible obtenerla y manipularla debido a la rutina _030_COMMUNICATION. En esta rutina es donde están presentes los bloques de la función MESSAGE con la que, como se ha explicado anteriormente, es posible poder leer los estados de las señales de campo y poder escribir sobre el estado de los actuadores y poder cambiar su estado según la programación del controlador principal.

Las rutinas a partir de _040_ a _130_ son las que establecen el comportamiento de los componentes del sistema. Se han creado mediante los tags de las diferentes UDT's y de esta manera poder ahorrar programación, debido a que no se debe de programar todos los elementos uno a uno, si no que basta con asociarlos a su UDT determinada. Por último, en la rutina _140_ se define el comportamiento de los sistemas.

Seguidamente podemos apreciar los tres siguientes programas, todos ellos con la misma estructura, uno para cada pareja de células. Poseen las mismas rutinas, una primera y principal _000_MAIN_ROUTINE que se va ejecutando a cada ciclo de programa y que va llamando al resto de rutinas del programa.

El resto de rutinas son referentes a todos los subprocesos en los que se divide el proceso principal de las FAS. Estos serán comentados con mayor profundidad más adelante en el momento de comentar las secuencias de automático del sistema. Como se observa en todos los programas, las dos últimas rutinas son pertenecientes a la ejecución de las condiciones principales de cada FAS, haciendo que vuelvan todos los actuadores y manipuladores del sistema en cuestión a su posición inicial de reposo.

Cabe destacar que en la figura 3.16. el símbolo que se observa al lado izquierdo del nombre de la rutina define el lenguaje de programación con el que se ha creado la rutina. El programa principal (MAIN_PROGRAM) ha sido íntegramente programado en *ladder* ya que se trata de la programación en manual de los componentes. Por otro lado, los demás programas, vemos que poseen otro tipo de icono, es el perteneciente al SFC (*Secuencial Flow Chart*) que se ha utilizado para poder llevar a cabo las secuencias del modo automático.

Se ha decidido hacer esta organización de los programas para facilitar la visión de la estructura del programa, además nos permite poder distinguir claramente la programación de las UDT's, esto nos permitirá poder acceder a ellos con mucha más facilidad que en otras ocasiones.

3.6.2. Definición del tipo de datos

El tipo de datos empleados en la programación del controlador son los siguientes:

- **Booleans (BOOL):** son la mayoría de los *tags* presentes en el sistema. Todos los estados, ordenes y señales de entrada y salida son de tipo dicreto, pueden estar desactivada '0' o activadas '1'.

F202_VS01.C_TAKE	0	Decimal	BOOL	Ventosas brazo manipulador Orden de coger o...
F202_VS01.C_LEAVE	0	Decimal	BOOL	Ventosas brazo manipulador Orden de dejar o...
F202_VS01.C_TAKE_A	0	Decimal	BOOL	Ventosas brazo manipulador Señal automática...
F202_VS01.C_LEAVE_A	0	Decimal	BOOL	Ventosas brazo manipulador Señal automática...
F202_VS01.C_RESET_ALARM	0	Decimal	BOOL	Ventosas brazo manipulador Orden de reset d...

Fig.- 3.17. Ejemplo de tags "bool" del sistema. Fuente: (propia).

- **Real (REAL):** son muy pocos los *tags* del sistema que pertenecen a estos tipos de datos. Se suelen emplear cuando queremos expresar una cantidad, por ejemplo, el numero de bases introducidos al sistema o el nombre de rodamientos rechazado. En la figura 3.18. observamos las variables de la FAS 204 donde se almacenan los datos recibidos del potenciómetro que mide la altura del rodamiento. Donde, F204_AS01.XMINS es la medida realizada y F204_AS01.MINS es el *tag* interno que tratamos en el SCADA.

F204_AS01	{...}	{...}	ADET	Detector analógico medición rodamientos
F204_AS01.MINS	0.0	Float	REAL	Detector analógico medición rodamientos Med...
F204_AS01.XMINS	0.0	Float	REAL	Detector analógico medición rodamientos Señ...

Fig.- 3.18. Tags tipo real. Fuente: (propia).

- **UDT's (User Data Type):** como se ha explicado anteriormente este tipo de datos se utilizan para minimizar la programación y la cantidad de línea de código, ya que agrupando los elemento que tiene el mismo comportamiento, se programan una sola vez en una variable de este tipo, con las señales creadas por el usuario. Una vez programado el control, solamente se deben instanciar las variables a la rutina de comportamiento como veremos a continuación.

En la figura 3.19. se puede apreciar a la izquierda el *tag* de los elementos del sistema y a su derecha el tipo de UDT a la que pertenece.

Data Types		Name	Value	Force Mask	Style	Data Type
		+ F207_CY01	{...}	{...}		CYS0
		+ F207_CY02	{...}	{...}		CYS1
		+ F207_CY03	{...}	{...}		CYS1
		+ F207_CY04	{...}	{...}		CYS0
		+ F207_JS01	{...}	{...}		DET
		+ F207_JS02	{...}	{...}		DET
		+ F208_CY01	{...}	{...}		CYS2
		+ F208_CY02	{...}	{...}		CYD2
		+ F208_CY03	{...}	{...}		CYS2
		+ F208_JS01	{...}	{...}		DET
		+ F208_JS02	{...}	{...}		DET
		+ F208_RA01	{...}	{...}		RA2D
		+ F208_VS01	{...}	{...}		SC

Fig.- 3.19. UDT's del sistema (izquierda), elementos del sistema asociados a sus UDT's (derecha). Fuente: (propia).

Para realizar las instancias de una UDT a un elemento en particular se debe seguir el procedimiento siguiente:

- 1) Una vez creadas las UDT's se deben crear unas variables auxiliares las que debemos asociar a las UDT's creadas. (figura 3.20).

Name	Value	Force Mask	Style	Data Type
+ CYD2	{...}	{...}		CYD2
+ CYS0	{...}	{...}		CYS0
+ CYS1	{...}	{...}		CYS1
+ CYS2	{...}	{...}		CYS2
+ DET	{...}	{...}		DET

Fig.- 3.20. Variables auxiliares para la programaci3n del comportamiento de las UDT's. Fuente: (propia).

- 2) Al inicio de la programaci3n de la UDT debemos poner la funci3n de *Subroutine (SBR)* (figura 3.21) que hace que el elemento instanciado a esa rutina tome el nombre de la variable auxiliar y recorra toda la rutina para ejecutarse la programaci3n, por 3ltimo se debe incorporar en la 3ltima l3nea de programaci3n el *RETURN* para volver a la rutina principal del programa.

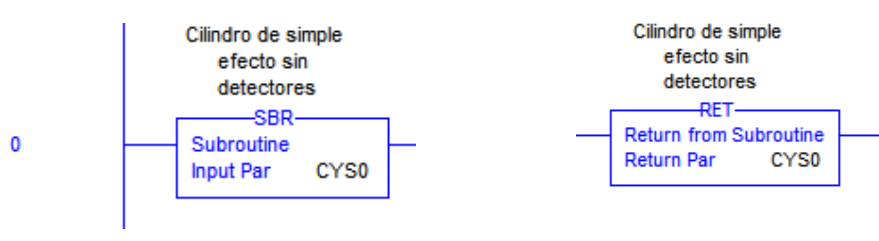


Fig.- 3.21. Uso de la variable auxiliar dentro de la rutina del UDT. Fuente: (propia).

- 3) Una vez creada la rutina de comportamiento mediante la variable auxiliar, debemos crear la llamada a la rutina de la UDT a partir de la *Main Routine*, que es la rutina que se va ejecutando constantemente a cada ciclo de programa. En la figura 3.22. observamos como hacemos la llamada a la subrutina de la UDT, incorporando el elemento real que queremos que recorra la programaci3n creada.

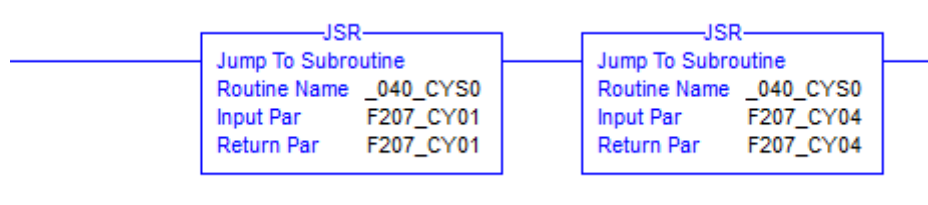


Fig.- 3.22. Llamada de la rutina de la UDT instanciando el elemento real que queremos que ejecute la rutina. Fuente: (propia).

Esto se aplica a todos los elementos del sistema con todas las rutinas de comportamiento de cada uno de ellos. La figura 3.23. muestra la *Main Routine* del programa principal donde se llama a todas las rutinas

instanciando a todos los elementos del sistema, de esta manera tiene su comportamiento definido y listas para operar en todo momento.

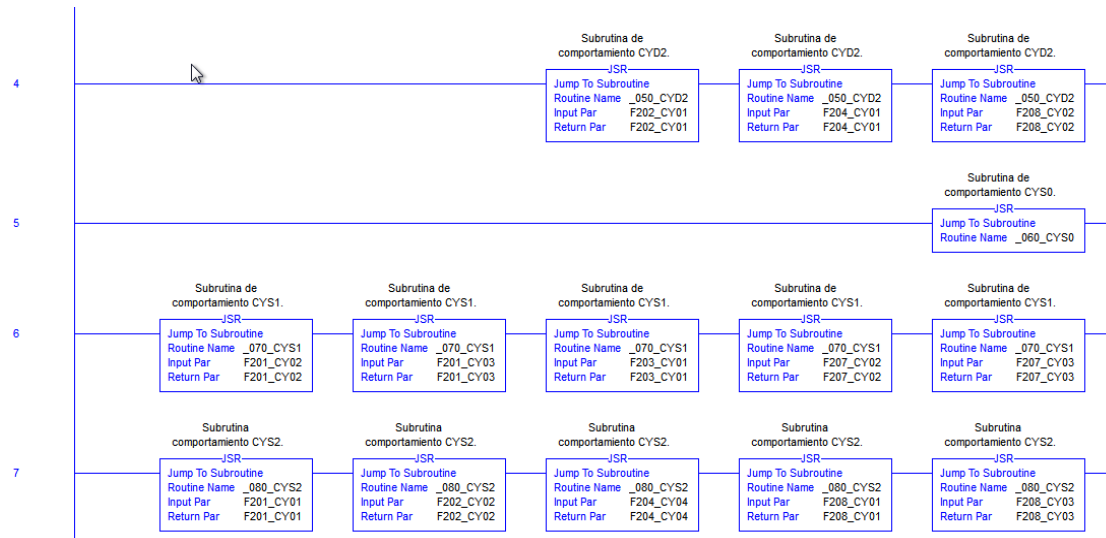


Fig.- 3.23. Main Routine del programa principal. Fuente: (propia).

- **Message:** este tipo de dato es utilizado en las comunicaciones con los controladores independientes de las FAS. Son los encargados de establecer la función *message* mediante la cual podemos establecer todos los parámetros para efectuar una comunicación eficiente.

+ MSG_X201	{ ... }	{ ... }	MESSAGE
+ MSG_X202	{ ... }	{ ... }	MESSAGE
+ MSG_X203	{ ... }	{ ... }	MESSAGE
+ MSG_X204	{ ... }	{ ... }	MESSAGE
+ MSG_X204_1	{ ... }	{ ... }	MESSAGE
+ MSG_X207	{ ... }	{ ... }	MESSAGE
+ MSG_X208	{ ... }	{ ... }	MESSAGE
+ MSG_Y201	{ ... }	{ ... }	MESSAGE
+ MSG_Y202	{ ... }	{ ... }	MESSAGE
+ MSG_Y203	{ ... }	{ ... }	MESSAGE
+ MSG_Y204	{ ... }	{ ... }	MESSAGE
+ MSG_Y207	{ ... }	{ ... }	MESSAGE
+ MSG_Y208	{ ... }	{ ... }	MESSAGE

Fig.- 3.24. Tags tipo message del sistema. Fuente: (propia).

- **Enteros (INT):** los *tags* con este tipo de dato (figura 3.25.) son utilizados como registros, para poder enlazar los registros de entrada y salida de los controladores de las FAS (*MicroLogix*) con las entradas y salidas creadas en el controlador principal (*CompactLogix*).

Estos datos poseen 16 posiciones discretas, por estos motivos podemos volcar el estado de los registros de los controladores independientes sobre ellos y manipularlos en nuestro programa. En la figura 3.11. del apartado x podemos ver la asignación de estos registros, los cuales, como se observa en la figura 3.26. y 3.27. enlazamos con los tags de entrada y salida de nuestro controlador principal.

+ XFAS201	0		Decimal	INT
+ XFAS202	0		Decimal	INT
+ XFAS203	0		Decimal	INT
+ XFAS204	0		Decimal	INT
+ XFAS204_1	0		Decimal	INT
+ XFAS207	0		Decimal	INT
+ XFAS208	0		Decimal	INT
+ YFAS201	0		Decimal	INT
+ YFAS202	0		Decimal	INT
+ YFAS203	0		Decimal	INT
+ YFAS204	0		Decimal	INT
+ YFAS207	0		Decimal	INT
+ YFAS208	0		Decimal	INT

Fig.- 3.25. Tags tipo INT. Fuente: (propia).

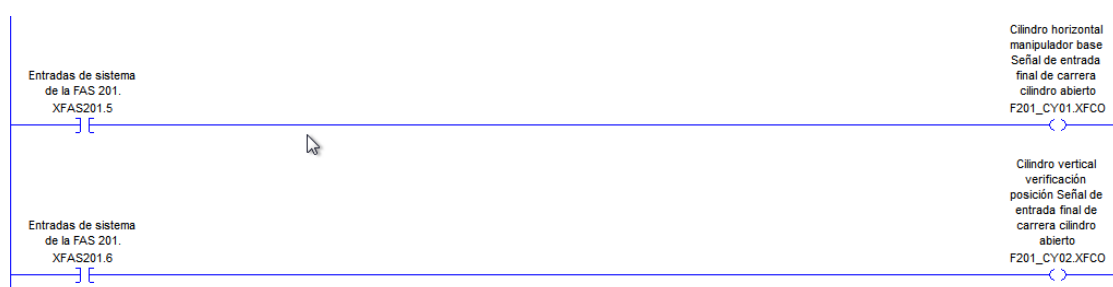


Fig.- 3.26. Enlace de los INT, asociados a las entradas físicas de las FAS mediante mensajería, con las entradas del sistema en el controlador principal. Fuente: (propia).

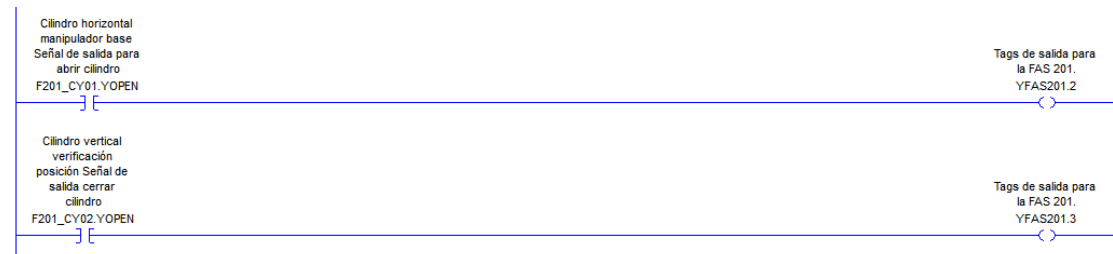


Fig.- 3.27. Enlace de los INT, asociados a las salidas físicas de las FAS mediante mensajería, con las salidas del sistema en el controlador principal. Fuente: (propia).

- Timer: son los tipos de datos utilizados para crear los temporizadores del sistema, que nos han servido para la activación de alarmas y conteo de tiempo para activar secuencias.

+ TIMER_COMMS	{ ... }	{ ... }		TIMER
+ TIMER_CY12	{ ... }	{ ... }		TIMER
+ TIMER_CY34	{ ... }	{ ... }		TIMER
+ TIMER_CY78	{ ... }	{ ... }		TIMER
+ TIMER_F201_BASE_POSITION	{ ... }	{ ... }		TIMER

Fig.- 3.28. Variables tipo TIMER. Fuente: (propia).

- SFC Action y SFC Step: estos tipos de datos son creado automáticamente por el *software* en el momento de crear las rutinas de lengua SFC. A medida que se van incorporando acciones y pasos en la secuencia, estos se van creando. Los *STEP* son los pasos que se deben ir ejecutando durante el proceso de la rutina y las *ACTION* son las acciones que se van realizando a medida que se van ejecutando las diferentes *STEP*.

+ Action_095	{...}	{...}		SFC_ACTION
+ Action_096	{...}	{...}		SFC_ACTION
+ Action_097	{...}	{...}		SFC_ACTION
+ Action_098	{...}	{...}		SFC_ACTION
+ Action_099	{...}	{...}		SFC_ACTION
+ Step_000	{...}	{...}		SFC_STEP
+ Step_001	{...}	{...}		SFC_STEP
+ Step_002	{...}	{...}		SFC_STEP
+ Step_003	{...}	{...}		SFC_STEP
+ Step_004	{...}	{...}		SFC_STEP
+ Step_005	{...}	{...}		SFC_STEP

Fig.- 3.29. Tipos de datos asociados al lenguaje SFC. Fuente: (propia).

3.6.3. Lógica de control de los elementos del sistema

Como se ha comentado anteriormente, en el *MAIN PROGRAM* se han creado las rutinas de comportamiento de los elementos, junto con las UDT correspondiente a cada uno de ellos. Los componentes del sistema los podemos agrupar en dos tipos: los controlados por válvulas neumáticas y los detectores del sistema. Dentro de los primeros hay una pequeña diferencia, se trata de las posiciones que tiene la válvula que controla el actuador. Si la válvula posee una posición son de un tipo, si la válvula posee dos posiciones o más son de otro tipo. Se han agrupado de esta manera debido a su comportamiento similar, que se refleja en una programación pareja en cada uno de estos componentes.

Los componentes pertenecientes al grupo controlado por válvulas neumáticas con una sola posición son los que poseen los UDT siguientes:

- CYS0
- CYS1
- CYS2
- RA2D

Por otro lado, los componentes que posee dos o más posiciones en su válvula son los que poseen los UDT siguientes:

- CYD2
- GRAP
- RA3D
- SC

Finalmente, quedan los elementos restantes que son los que pertenecen al grupo de detectores del sistema, son básicamente todos los sensores, tanto discretos como analógicos y son los que pertenecen a las UDT's siguientes:

- DET
- ADET

Como se podrá comprobar, todas las rutinas de comportamiento de los componentes poseen una estructura similar para facilitar la navegación y la búsqueda de elementos, errores, etc.

Elementos controlados por válvulas neumáticas

Para analizar la programación de este tipo de elementos y como tienen un comportamiento similar, se va a escoger la rutina de comportamiento de los cilindros de doble efecto con dos detectores de final de carrera, abierto y cerrado (CYD2) para los componentes que cuentan con dos o más posiciones de válvula. Para los que poseen una sola posición de válvula se utiliza la rutina del cilindro de simple efecto con un detector (CYS1).

Los cilindros y los actuadores rotativos tienen un comportamiento idéntico y además poseen los mismos sensores de final de carrera. La única diferencia con las ventosas es que estas no poseen sensores de final de carrera, sino que poseen un presostato. Este sensor es parecido a un final de carrera, la activación de este sensor nos notifica de que la ventosa ha alcanzado la presión correcta para poder mover el objeto a la nueva posición. Este sensor se trata de igual manera que un final de carrera de un cilindro o un actuador rotativo.

Por último, tenemos las pinzas neumáticas que no poseen sensores, por lo demás el comportamiento y activación de sus válvulas es idéntica que a los demás componentes.

La asignación de las UDT's a estos componentes se ha llevado a cabo teniendo en cuenta los sensores que poseen y el tipo de válvula que lo controla, como hemos hecho en la separación que se ha efectuado anteriormente. Si la válvula es monoestable el cilindro será de simple efecto, por otro lado, si la válvula es biestable el cilindro deberá ser de doble efecto ya que esto va regulado según las posiciones que posee la válvula que controla el elemento.

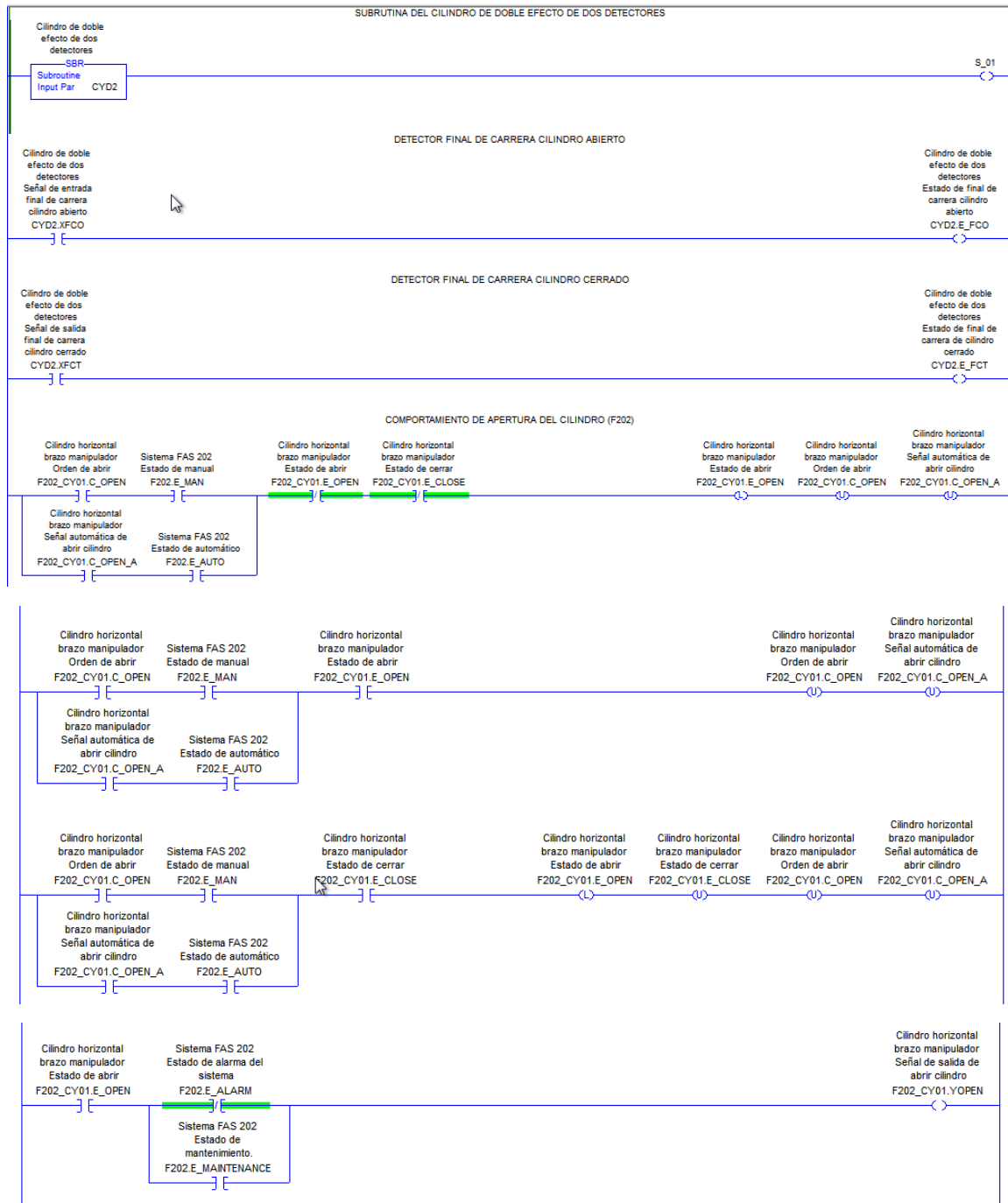


Fig.- 3.30. Ejemplo de lógica de control de apertura con dos o más posiciones de válvula con UDT CYD2.

Fuente: (propia).

En primer lugar y como podemos apreciar en la figura 3.30., lo primero que se hace en la rutina es introducir la variable auxiliar a través de la cual se va a instanciar al elemento del sistema. Los siguiente es definir que en el momento que un sensor se active en el sistema, el controlador lo debe de registrar como un estado de dicho sensor.

Seguidamente se define el comportamiento de apertura, en este caso del cilindro, pero podría ser de cualquier elemento del grupo de elementos controlador por válvulas de dos posiciones o más.

Para activar el estado de apertura del cilindro el sistema debe de recibir la orden pertinente, indiferente si es de manera manual o automática pero el estado de funcionamiento debe de estar activado para cada uno de estos casos. Cumpliendo estas condiciones se provoca un *latch* en el estado de *OPEN* del cilindro. Esto provoca una activación mantenida del estado, aunque ya no se cumplan las condiciones de activación. Además, se provoca un *unlatch* (desactivación mantenida) de las ordenes de apertura para que no se queden activadas.

Como podemos observar en la parte inferior si se recibe la orden de abrir el cilindro y el estado ya esta activado, no ocurre nada, solamente se provoca el *unlatch* de las ordenes de apertura.

Por otro lado, si se recibe la orden de apertura y esta activado el estado de cerrado, este último estado se desactiva y como vemos, por medio del *latch*, se activa el estado de apertura.

Más abajo se comprueba como el estado de apertura provoca una activación de la salida que fuerza el sistema a activar físicamente el cilindro.

Cabe destacar que la activación del estado de activación de los sensores no se provoca mediante la activación mantenida (*latch*) debido a que nos interesa que este activado solamente en el momento que físicamente el sensor este activo en la célula.

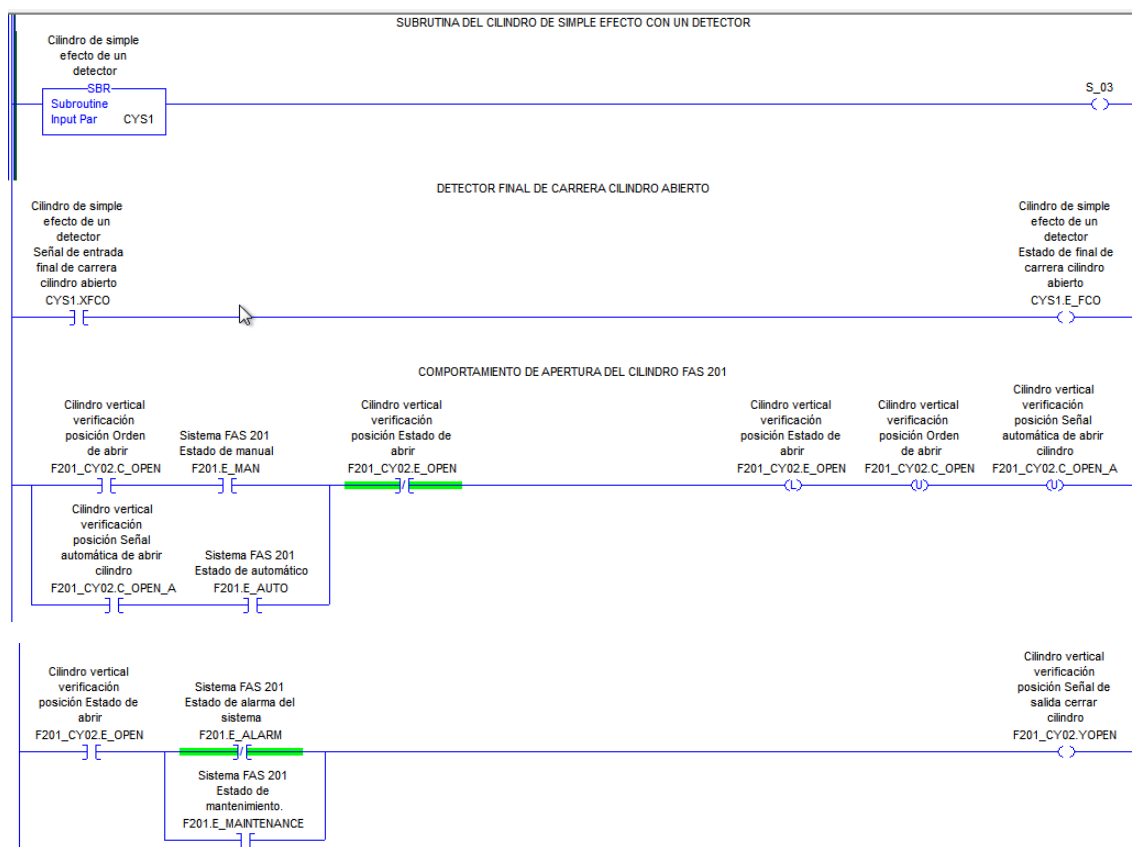


Fig.- 3.31. Ejemplo de lógica de control de apertura con una posición de válvula con UDT CYS1. Fuente: (propia).

En este caso se hace lo mismo, llamar a la variable auxiliar y definir la activación del sensor final de carrera. Por último, se provoca un *latch* al estado de apertura. Aquí no encontramos el estado de cierre ya que la válvula solo posee una sola posición.

Por último, vemos que el estado de apertura provoca la activación de la salida, justo como en el caso del cilindro de doble efecto, comentado anteriormente.

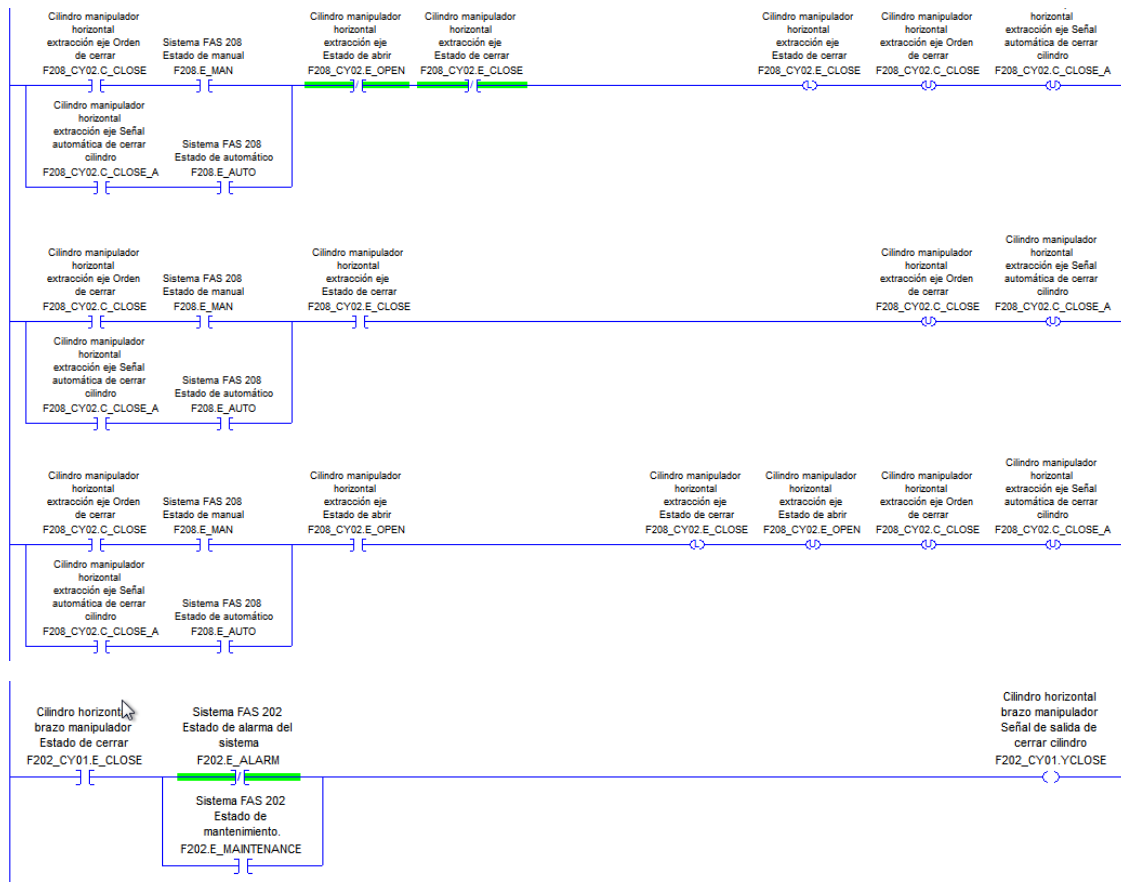


Fig.- 3.32. Ejemplo de lógica de control de cierre con dos o más posiciones de válvula con UDT CYD2.

Fuente: (propia).

En la figura 3.32. apreciamos la lógica de cierre de los componentes de dos posiciones o más de válvulas, se comporta exactamente igual que la apertura, solo cambiando la orden, el estado y la salida.

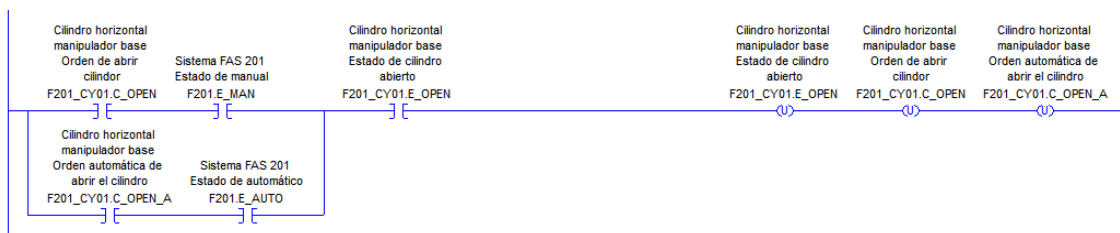


Fig.- 3.33. Ejemplo de lógica de control de cierre con una posición de válvula con UDT CYS1. Fuente:

(propia).

Para los componentes de solo una posición de válvula, para que la orden de cierre se haga efectiva, simplemente se debe desactivar el estado de apertura que activa la salida, para ello se debe de pulsar de nuevo la orden de apertura cuando el estado de apertura este activado (figura 3.33.).

A continuación, se procede a enseñar la parte de alarmas, que es común para todos los elementos, indiferentemente de las posiciones de válvula que tengan.

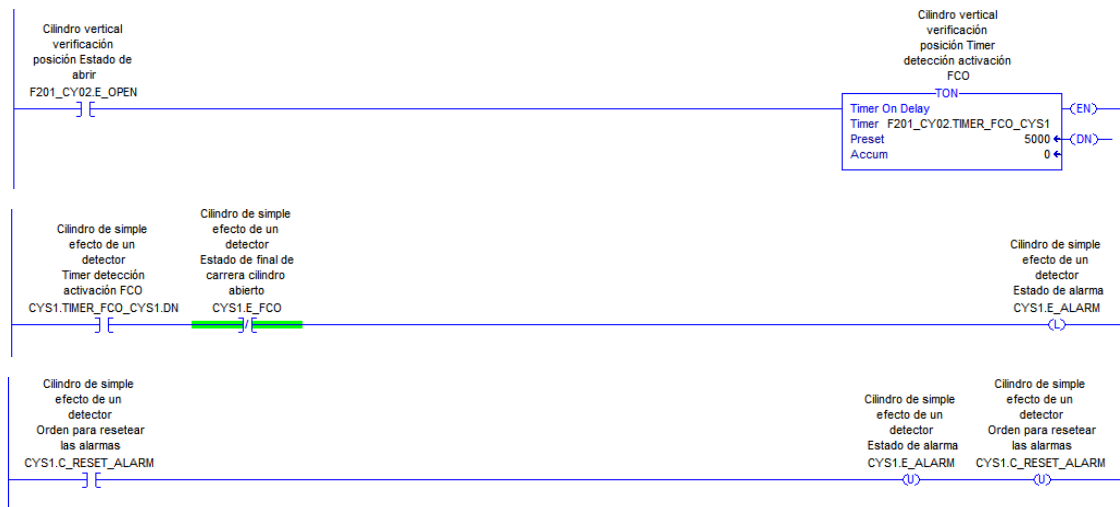


Fig.- 3.34. Ejemplo de lógica de control de cierre con una posición de válvula con UDT CYS1. Fuente: (propia).

Como vemos en la figura 3.34., en este caso es la apertura de un cilindro, pero puede ser la activación de cualquier estado que contenga al menos un sensor aparejado, en el momento de activar dicho estado se activa un *timer* que controla el tiempo que el elemento físico tarda hasta llegar a su final de carrera, posición final o presostato activado, dependiendo del componente que se este tratando. Si por alguna anomalía del sistema el sensor asociado a dicha activación no se activase en el tiempo determinado, como vemos en la segunda línea de código, la alarma se activa, indicando un mal funcionamiento del actuador. En la última línea se ejecuta el reseteo de la alarma que se ha activado, siempre y cuando la condición que ha activado la alarma ya este deshabilitada, en caso contrario, la alarma seguirá generándose.

Elementos detectores

Estos elementos como se ha comentado previamente pueden ser de dos UDT diferentes, DET y ADET.

La ADET está referenciada a los sensores analógicos, en el sistema solo encontramos el potenciómetro presente en la FAS 204 que sirve para poder medir la altura del rodamiento y poder hacer la selección entre altos y bajos.

En la figura 3.35. podemos ver el comportamiento de esta UDT, se debe simplemente ejecutar el comando *MOVE*, para mover del *tag* de entrada el valor al *tag* que este asociado con el SCADA y que permite visualizar el valor de la variable en la aplicación.



Fig.- 3.35. UDT ADET. Fuente: (propia).

Por último, la figura 3.36. muestra la UDT DET, que simplemente activa el estado de activación del sensor en el momento que la señal física esté activada.

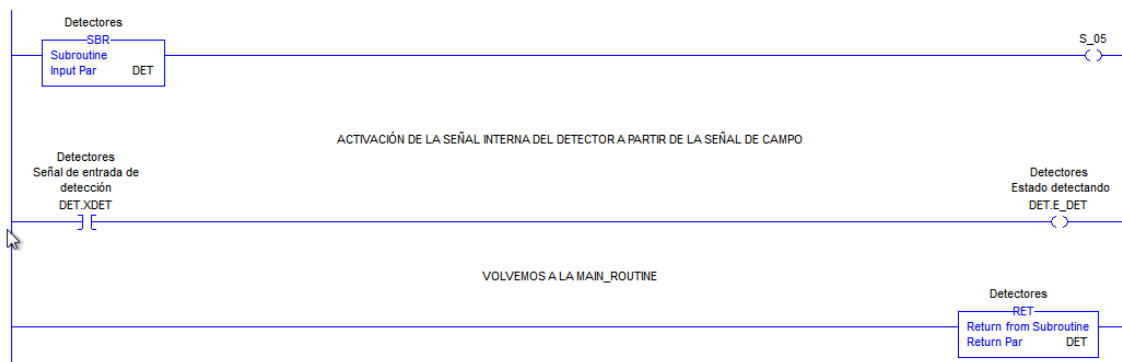


Fig.- 3.36. UDT DET. Fuente: (propia).

Como en todas las rutinas de comportamiento de los elementos, vemos que la primera orden es la definición de la variable auxiliar para poder instanciar el elemento que deseamos.

3.6.4. Lógica de control de los sistemas

Se procede a explicar con más detalle la rutina `_140_SYS` del programa *MAIN PROGRAM*.

Para poder entrar en el modo remoto del sistema se debe pulsar al mismo tiempo los botones de *START* y *STOP* de la botonera de la FAS y que el selector de modo este en *AUTO*. Una vez entrado en remoto para poder controlar el sistema, se enciende el led azul de la FAS.

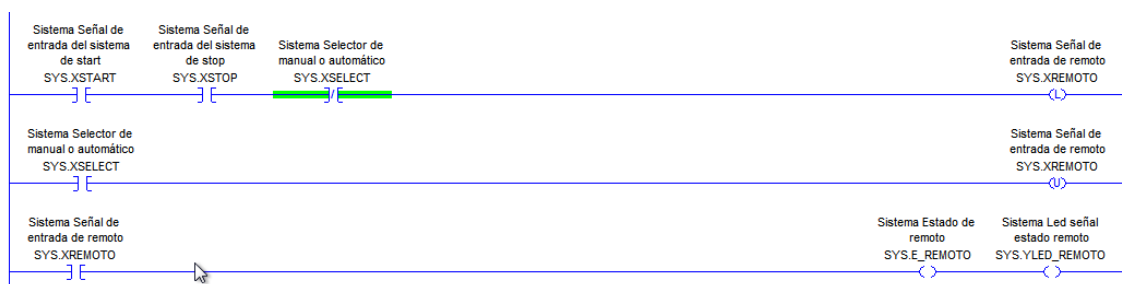


Fig.- 3.37. Entrada en remoto del sistema. Fuente: (propia).

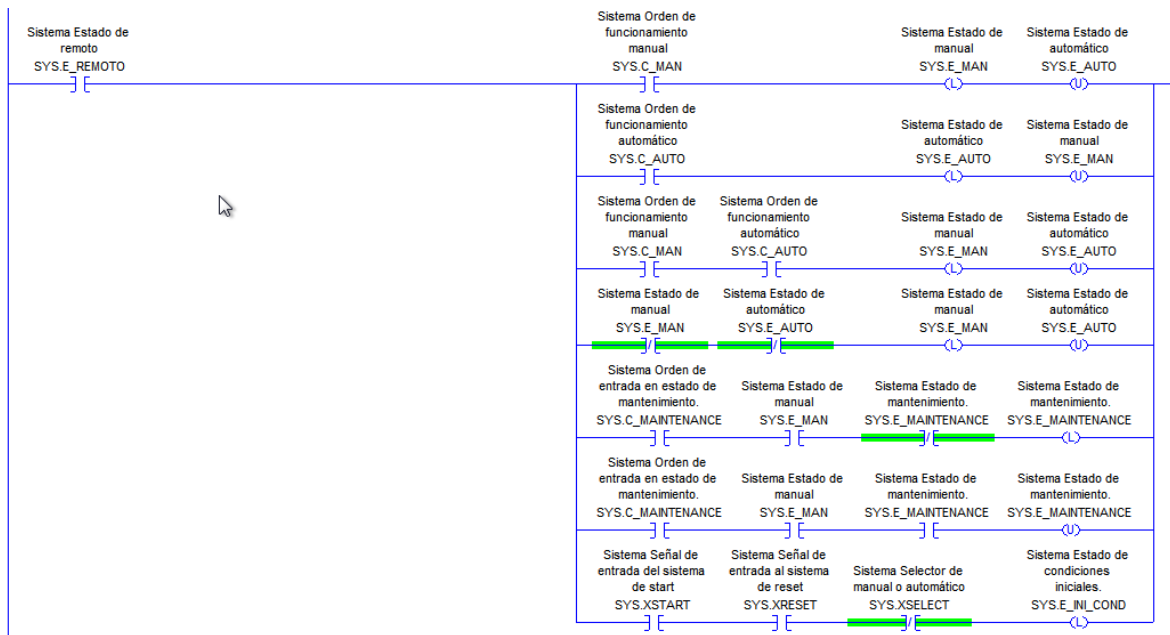


Fig.- 3.38. Comportamiento de los modos del sistema. Fuente: (solución).

Una vez en el estado de remoto si no hay ninguna orden del modo en el que debe funcionar, el sistema pasará a manual directamente, si se da la orden al mismo tiempo de manual como de automático el sistema se mantiene en manual.

Los estados son complementarios, la activación de uno desactiva la contrario y viceversa. Como se puede apreciar en la figura 3.38. el sistema cuenta con un modo de mantenimiento, que solo se puede acceder cuando el sistema esta en manual. Este modo de mantenimiento nos permite poder efectuar cualquier orden y movimiento de actuadores del sistema, aunque haya activas alarmas. Esto nos sirve para poder desactivar las alarmas activas.

También cuenta con un estado de condiciones iniciales, a este estado se puede acceder mediante la botonera de la FAS, pulsando simultáneamente los botones de *START* y *RESET*. Esto inicia una secuencia que hace que todos los actuadores, del sistema que se haya activado, vuelven a su posición inicial.

Si en algún momento el selector de modo de la botonera de la FAS pasa a *MAN* el sistema saldría de remoto y no se podría controlar el modo de funcionamiento desde la aplicación SCADA hasta que no se volviera a entrar en remoto mediante la botonera.

En la figura 3.39. se observa el reseteo de las alarmas de sistema de cada FAS. Para que se hagan efectivas, las alarmas de cada elemento de la FAS en cuestión no deben de estar activas, ya que son ellas las que con su activación también activan las del sistema en el que se encuentre dicho elemento que entra en alarma.

Se dispone de dos botones que nos permiten, siempre y cuando todas las células se encuentren en remoto, pasar a estado manual o automático directamente todas las células al mismo tiempo.

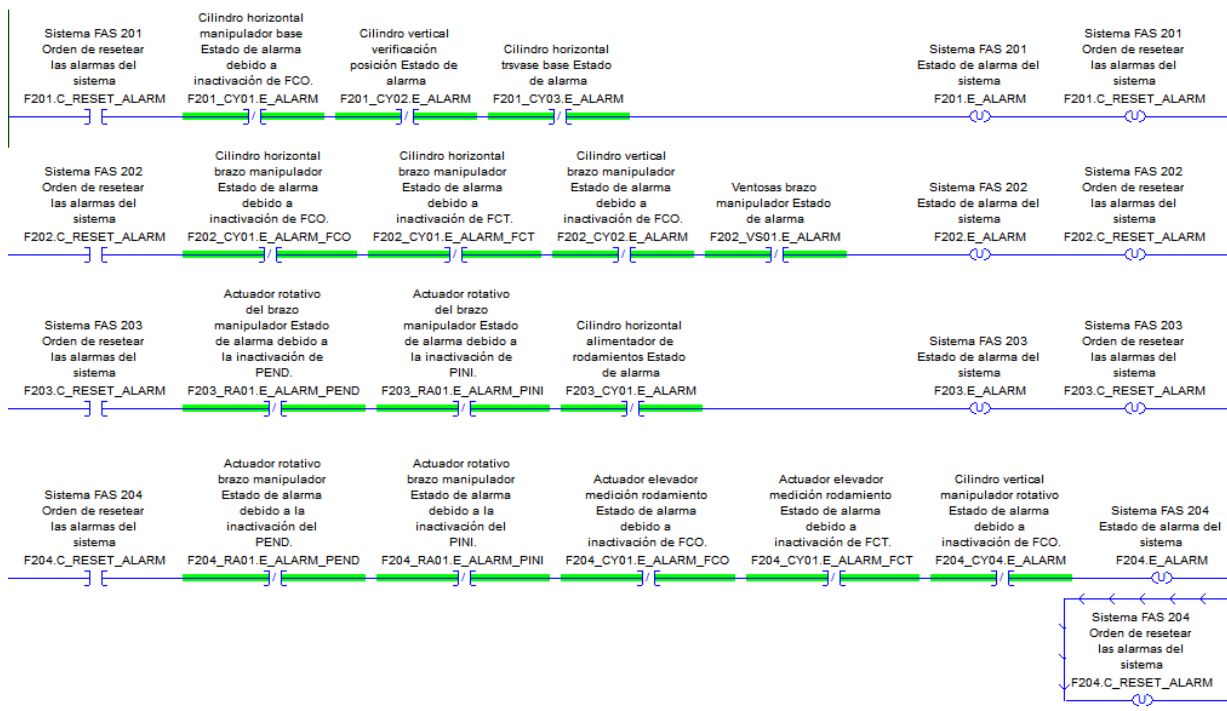


Fig.- 3.39. Reseteo de las alarmas de sistema. Fuente: (propia).

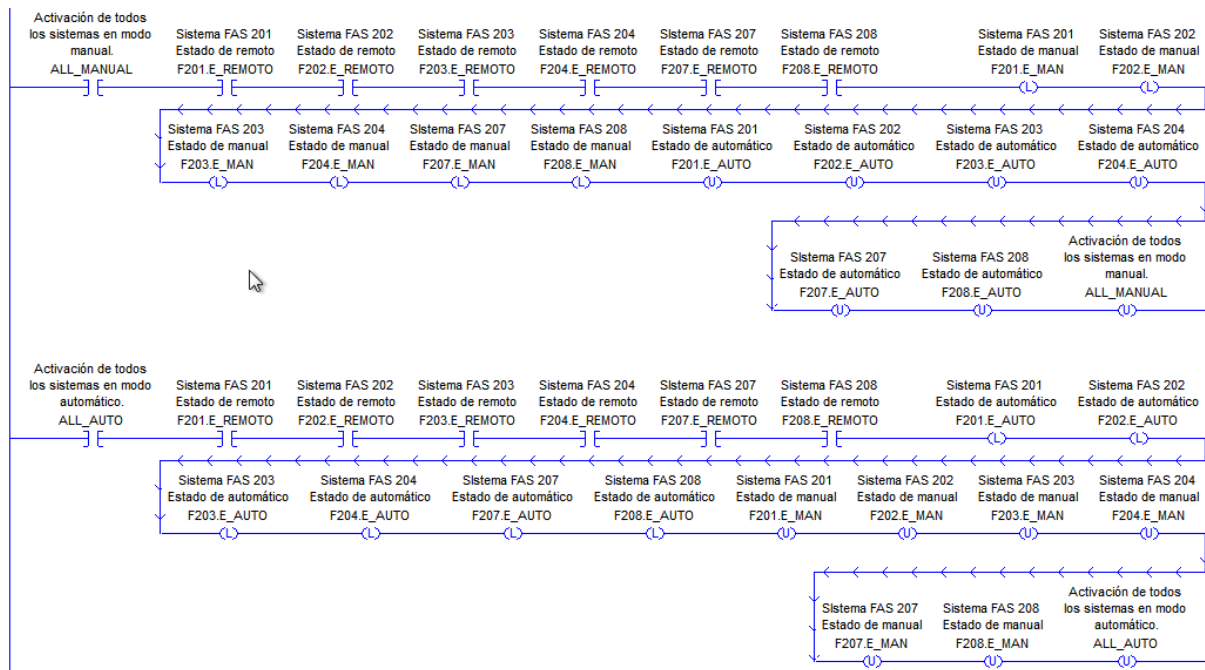


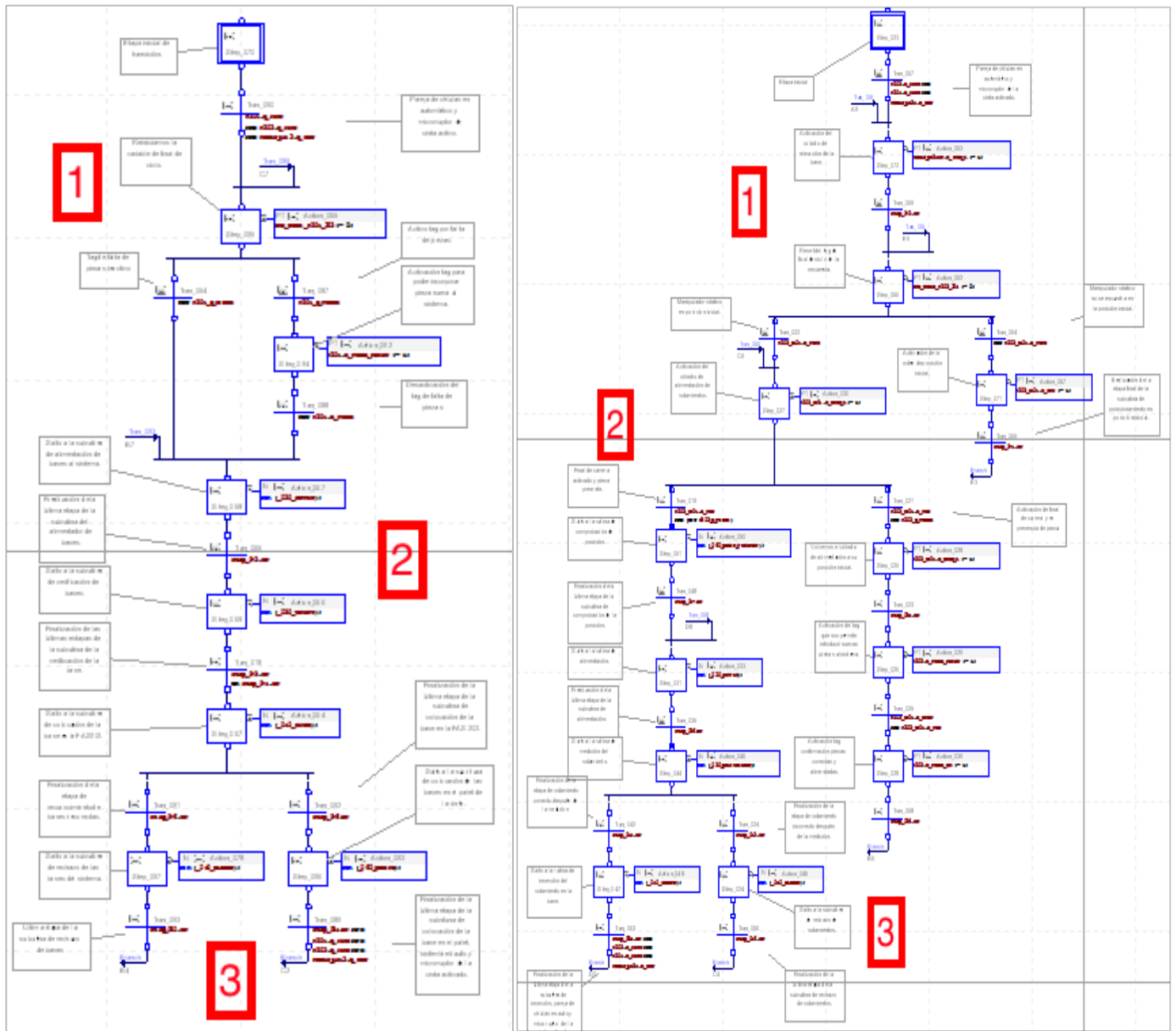
Fig.- 3.40. Cambio de modo general. Fuente: (propia).

3.6.5. Secuencias de control

La secuencia de control es la que controla el funcionamiento del modo autom1tico y como el sistema se va ejecutando y siguiendo una pauta de manera ordenada.

Como el sistema general a automatizar consta de 3 parejas de c3lulas, se dispone de 3 secuencias principales que rigen el comportamiento aut3mata de las c3lulas.

En las siguientes tres figuras se observan las 3 secuencias de manera ordenada y con etapas diferencias para que sea m1s f1cil y eficiente la explicaci3n de su funcionamiento. En el anexo 1 pueden consultar los SFC's con mayor precisi3n.



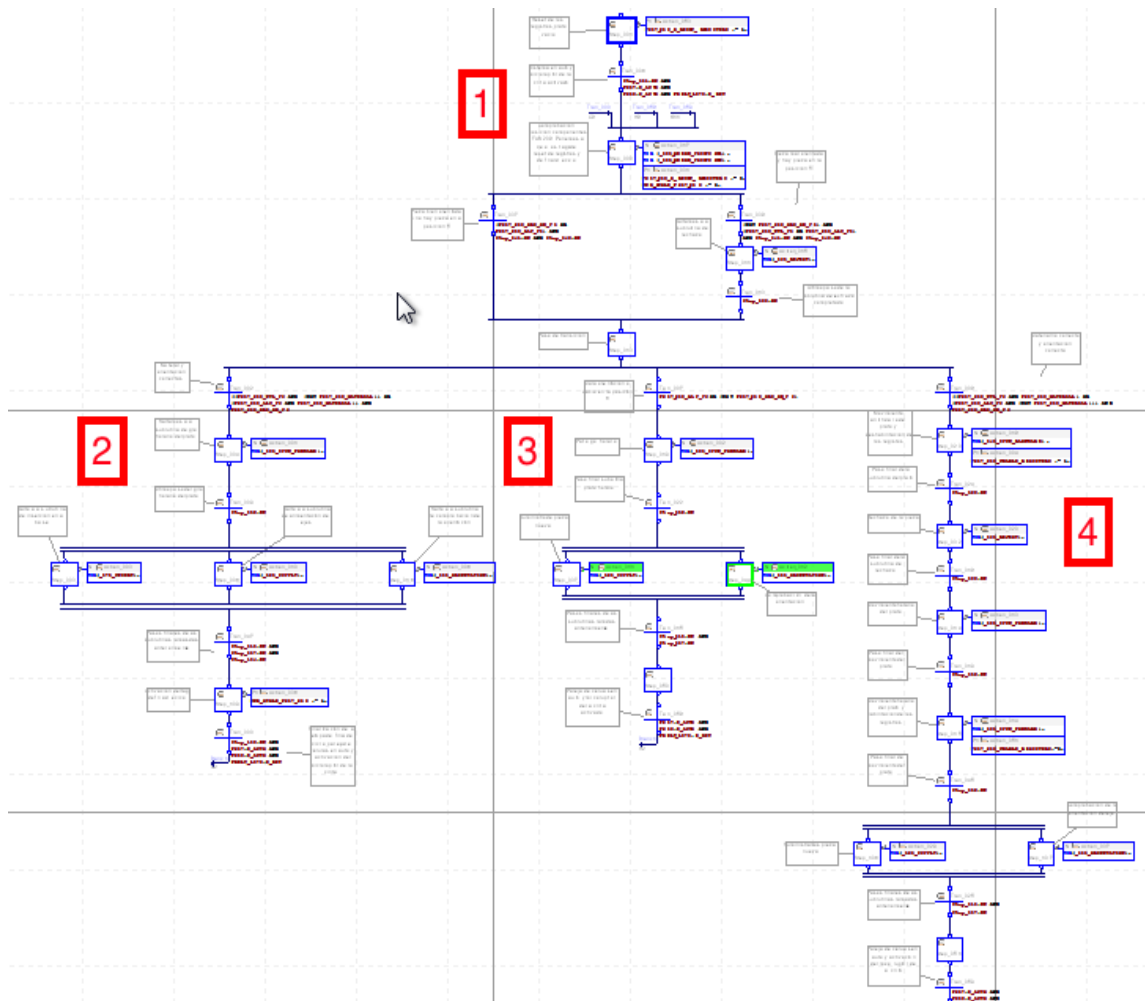


Fig.- 3.43. Secuencia de la pareja FAS207 y FAS208. Fuente: (propia).

FAS201 - FAS202

La secuencia general de las dos parejas se va desarrollando a lo largo que se van cumpliendo las condiciones para poder avanzar a la siguiente etapa del SFC general. Durante su ejecuci3n el SFC tiene *JSR (Jump To Subroutine)* con esto saltamos a otras subrutinas que realizan subprocesos de la pareja de FAS que se est3n ejecutando. Estos subprocesos son los que podemos ver m3s detalladamente en el apartado 2.2. donde se expone todos sus componentes f3sicos y su comportamiento.

En primer lugar, debemos saber que la primera pareja tiene la capacidad de comprobar la orientaci3n de la base que alimenta al sistema, de esta manera es capaz de rechazar y sacar fuera del circuito de ensamblaje a cualquier base con una orientaci3n err3nea.

En el momento que el sistema identifica una base correcta la procede a insertar en el *palet* que se encuentra en la cinta transportadora.

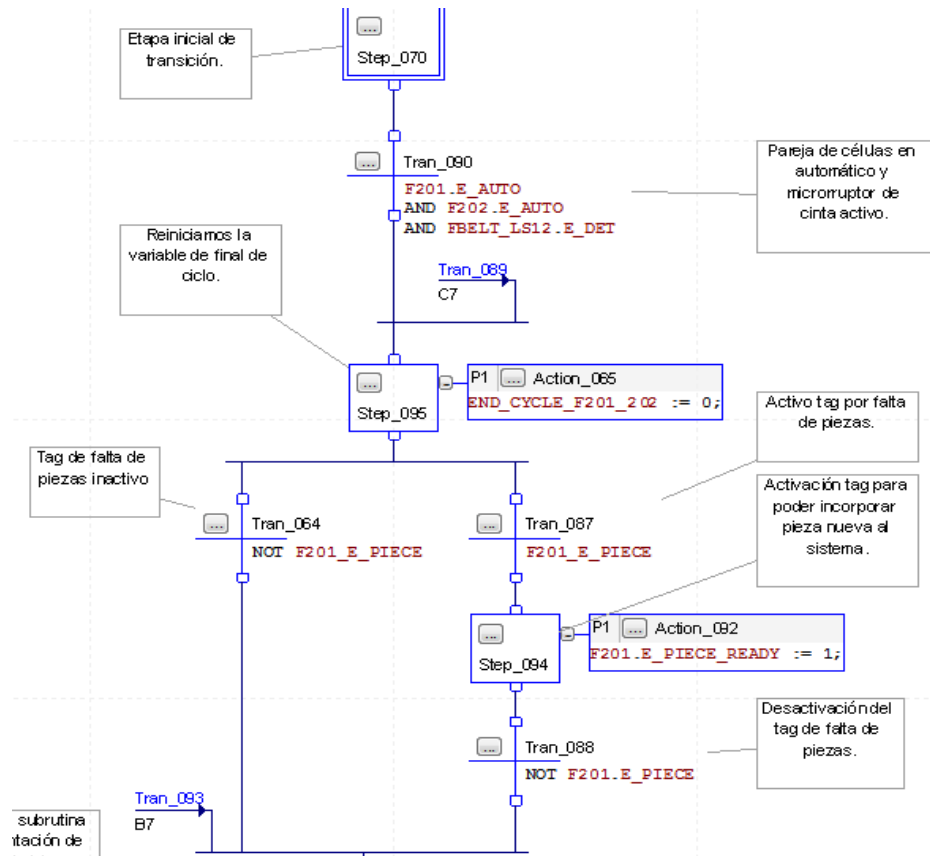


Fig.- 3.44. Parte 1 del SFC de la primera pareja. Fuente: (propia).

- 1) La secuencia (figura 3.44.) se inicia justo al arrancar el controlador porque la *MAIN ROUTINE* posee un *JSR (Jump To Subroutine)* y se queda enclavada en la etapa inicial, pero no entra en juego hasta el momento que no cumple la primera condición. Una vez las dos células se encuentren en automático y el microinterruptor de la cinta este activo, empieza la secuencia. Vemos que se resetea una variable de final de ciclo (*END_CYCLE_F201_2_Q2*), esta variable nos notifica cuando el ciclo total del SFC se ha terminado por este motivo cuando empieza un nuevo ciclo se resetea. Seguidamente, la secuencia se abre en dos caminos y nos da la casuística de si faltan bases en el sistema, si es así, seguirá el camino de la derecha. Esto activa un *tag (F201.E_PIECE)* que realiza la acción que podemos observar en la figura 3.45.



Fig.- 3.45. Activación del LED y del bit de falta de piezas. Fuente: (propia).

Observamos que activa el *LED* rojo de la FAS que indica la falta de piezas en el sistema. Por otro lado, también activa el *tag (F201.E_PIECE)* de falta de piezas en el sistema. Este *tag* se encarga de deshabilitar cualquier movimiento de los actuadores del sistema afectado, hasta el momento en el que este sea desactivado, como se aprecia en la figura 3.46.



Fig.- 3.46. Desactivación de los actuadores cuando en el sistema faltan piezas. Fuente: (propia).

En el *Step_094* del SFC se activa el tag (*F201.E_PIECE_READY*) este tag hace aparecer una pantalla de aviso en la aplicación SCADA mediante la que notifica al usuario que debe introducir bases al sistema. Una vez hecho esto el usuario debe pulsar el botón *OK* de la ventana emergente de la aplicación. El botón *OK* activa el tag (*F201.C_PIECE_OK*) que realiza la acción que se aprecia en la figura 3.47.

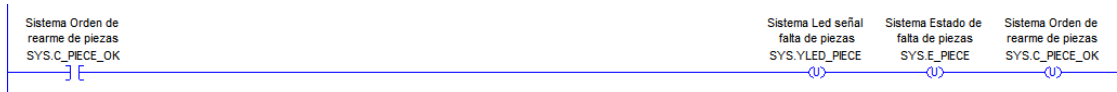


Fig.- 3.47. Activación del tag *F201.C_PIECE_OK*. Fuente: (propia).

Este tag desactiva el *LED* rojo de falta de piezas y el tag interno por falta de piezas que desactivaba los actuadores. Con esto queda habilitado el sistema de nuevo. Habiendo alimentado el sistema podemos seguir hacia la segunda parte del SFC. Por otro lado, el camino de la izquierda es el que identifica que todo está correcto y sigue hacia la segunda parte.

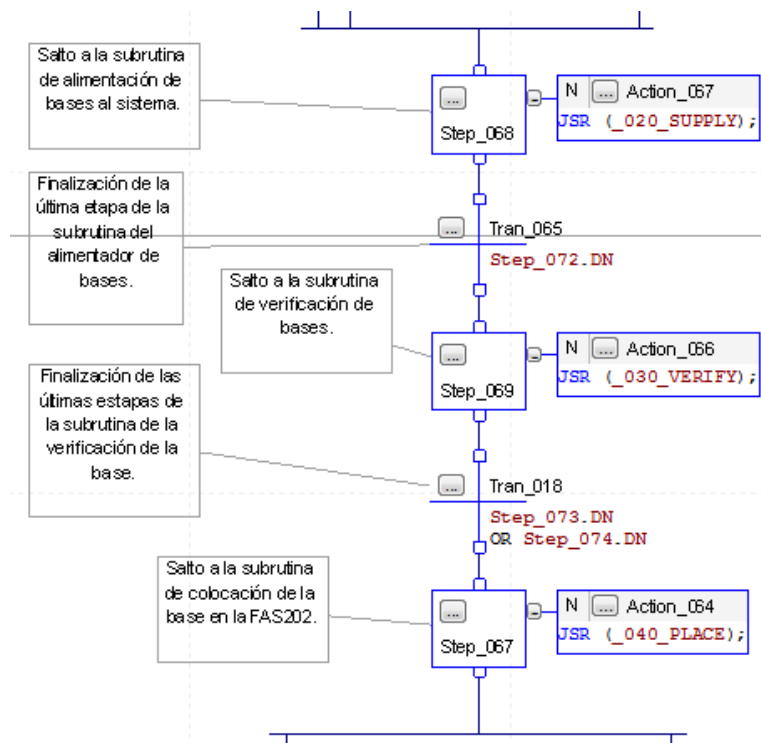


Fig.- 3.48. Parte 2 del SFC de la primera pareja. Fuente: (propia).

- 2) Siguiendo con la comprobación de la falta de piezas hecha, vemos que el programa (figura 3.48.) hace un salto a la subrutina de alimentación de bases al sistema. Una vez

que el *Step_072*, última etapa de la subrutina de alimentación, este terminada el programa vuelve al SFC principal. Seguidamente vuelve a saltar a una nueva subrutina, esta vez la de verificación de la orientación de la base. En el momento que se terminen, o bien *Step_073* o bien *Step_074* el sistema vuelve de nuevo a la secuencia principal. Seguidamente vuelve a saltar a una subrutina, en este caso la de transporte de la base de la FAS201 hacia la FAS202.

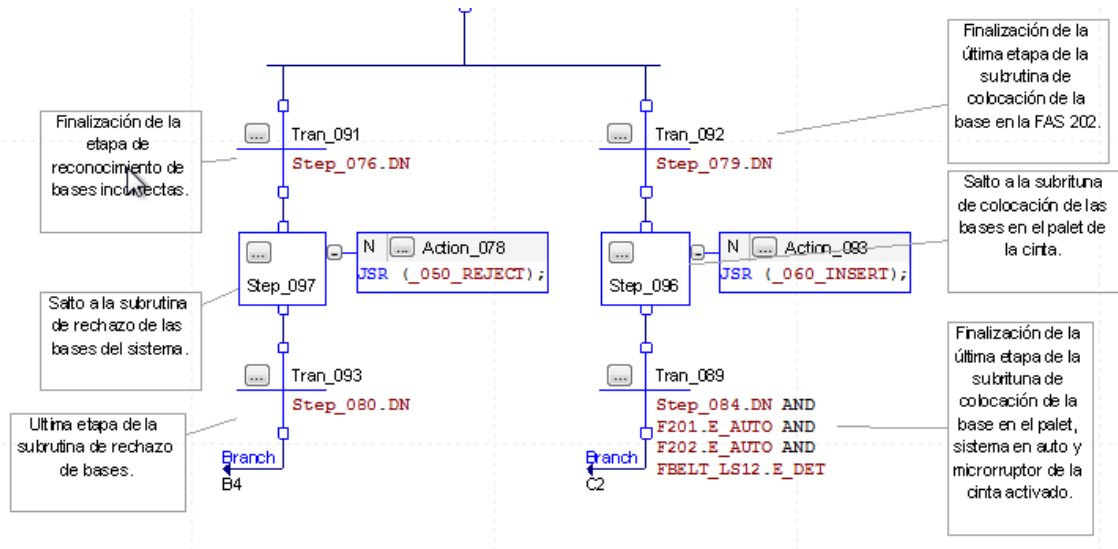


Fig.- 3.49. Parte 3 del SFC de la primera pareja. Fuente: (propia).

- 3) Una vez la secuencia ha saltado a la subrutina de transporte, dependiendo del *tag* generado en el momento de la comprobación de la orientación, el sistema hará una cosa u otra. Si se finaliza la *Step_076* (parte izquierda del SFC) es porque la pieza no estaba adecuadamente orientada, por lo tanto, el sistema salta a la subrutina de rechazo de bases. Una vez completada esta subrutina (activación del *Step_080*) se procede a volver al inicio de la segunda parte del SFC, donde se procede a volver a insertar una nueva base en el sistema.

En el caso contrario y se finaliza la *Step_079* la secuencia salta a la subrutina de insertar la base en el *palet* de la cinta. En el momento que esto se efectúe, el sistema queda a la espera de cumplir con la condición: ambas células en automático y microrruptor de la cinta activado, si esto se cumple, la secuencia vuelve al inicio del SFC, antes del paso de resetear la variable de final de ciclo y empieza de nuevo la secuencia.

FAS203 - FAS204

Al igual que la primera pareja, la segunda también posee en el interior de su SFC saltos a subrutinas de subprocesos, los podremos ver con mayor profundidad en el apartado 2.2.

La actual pareja tiene la capacidad de poder diferenciar entre rodamientos altos y bajos, mediante la medición de su altura en la FAS204. El usuario es el encargado de ordenar al sistema el tipo de rodamiento que desea, los que no coincidan con la orden del usuario serán expulsados del sistema, al igual que sucedía con las bases en la primera pareja.

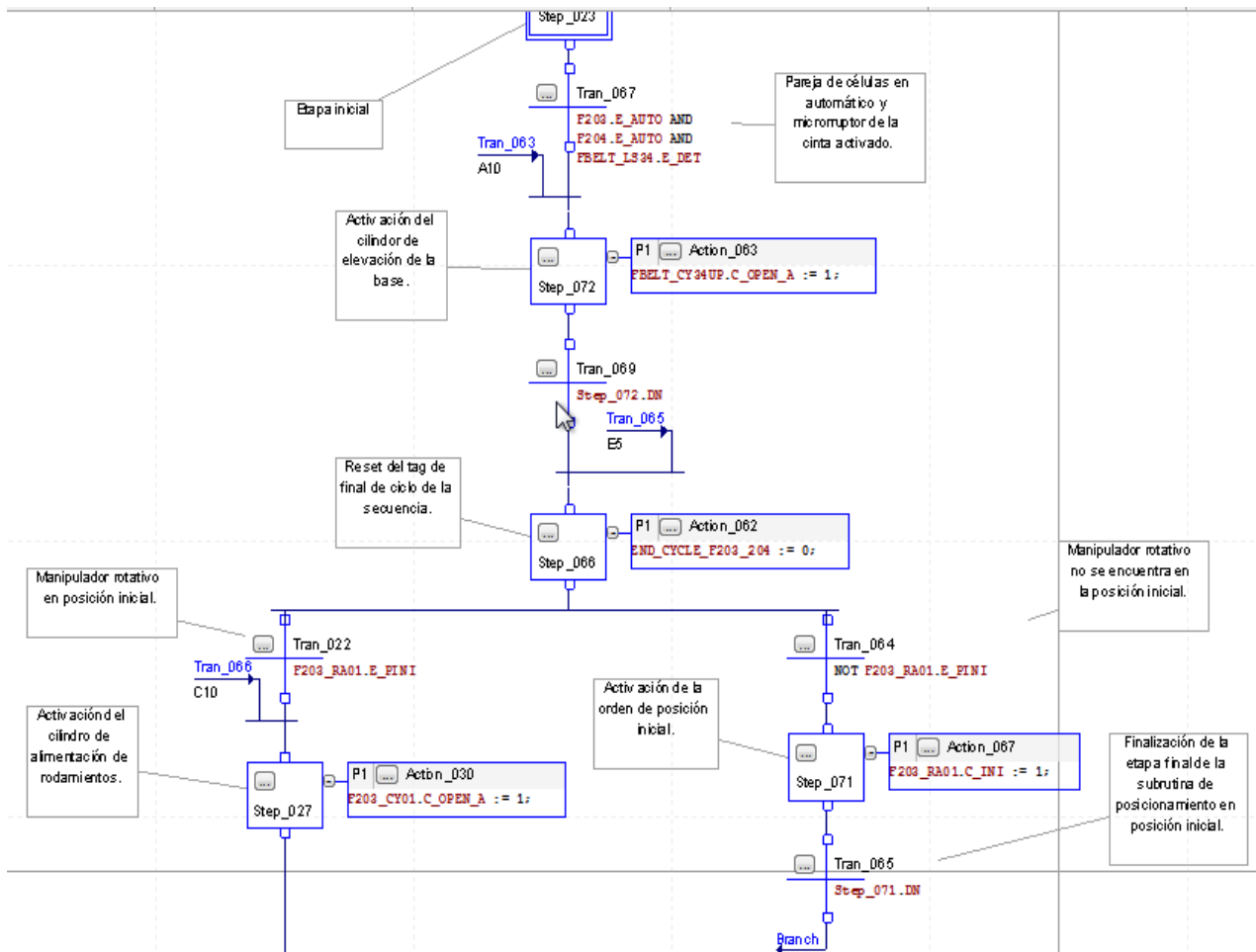


Fig.- 3.50. Parte 1 del SFC de la segunda pareja. Fuente: (propia).

- 1) Una vez las dos células se encuentren en automático y el microrruptor de la cinta este activado se da comienzo a la secuencia de la segunda pareja. La primera acción que se lleva a cabo es la elevación del cilindro donde se encuentra la base en la cinta, esto se debe hacer para facilitar la inserción del rodamiento posteriormente. Una vez esta acción este acabada, se resetea la variable de final de ciclo, al igual que en la primera pareja, se resetea cuando comienza una nueva secuencia y se activa en el momento que se finaliza. Seguidamente la secuencia controla si el actuador rotativo de la FAS203 se encuentra en posición inicial. Si se encuentra en posición inicial hacemos avanzar al cilindro de alimentación de rodamientos (parte izquierda del camino). Si, por el contrario, el actuador rotativo no se encuentra en la posición inicial (parte derecha del camino) forzamos la orden de posición inicial ($F203_RA01.C_INI$) y la secuencia vuelve al mismo punto a comprobar la posición de nuevo.

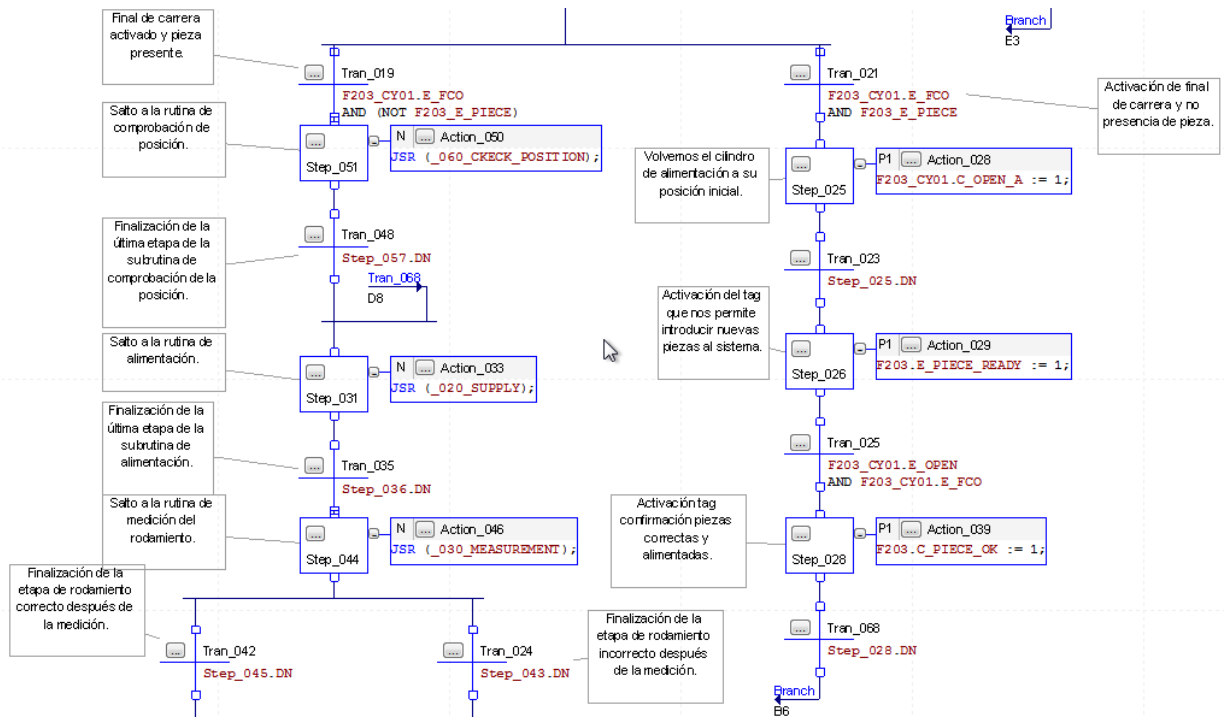


Fig.- 3.51. Parte 2 del SFC de la segunda pareja. Fuente: (propia).

- 2) La segunda parte sigue después de avanzar al cilindro de alimentación, una vez este llegue al final de carrera, se comprueba al mismo tiempo si se disponen de rodamiento o no. Esto se hace, mediante un microrruptor que se activa al llegar el cilindro al final de carrera y en el caso de que haya rodamiento. En el caso de que no hay rodamientos (parte derecha de la bifurcación), el microrruptor no se activa y la secuencia hace retroceder al cilindro y activa el tag ($F203_E_PIECE$), como sucede en la primera pareja también este tag activa el LED de falta de piezas de la FAS y activa el tag ($F203.E_PIECE$) como se aprecia en la figura 3.52.



Fig.- 3.52. Activación del tag de falta de piezas del sistema FAS203. Fuente: (propia).

Al mismo tiempo, el tag ($F203.E_PIECE$) desactiva el movimiento del actuador rotativo de la FAS203 (figura 3.53.).

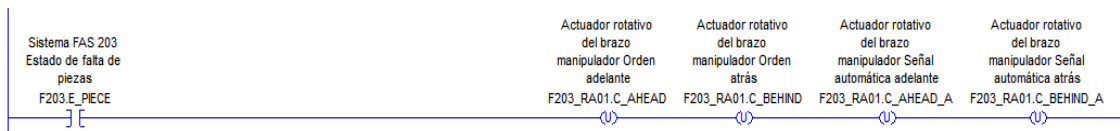


Fig.- 3.53. Desactivación del actuador rotativo por falta de rodamientos. Fuente: (propia).

Una vez retrocedido el cilindro, se activa un tag interno ($F203.E_PIECE_READY$) que hace aparecer en el SCADA un *pop-up* comunicando la falta de piezas y pidiendo que se

introduzcan más piezas en el sistema. Una vez hayamos alimentado al sistema con más rodamientos, podemos pulsar *OK* en la pantalla emergente. Este botón *OK* va a poner en movimiento el cilindro de alimentación para introducir de nuevo un rodamiento al sistema. Cuando el cilindro llegue a su final de carrera, se activa el *tag* que confirma que la pieza está *OK* (*F203.C_PIECE_OK*) este *tag* también desactiva la luz de la célula de emergencia por falta de piezas y también desactiva el *bit* de falta de piezas, permitiendo al actuador rotativo poder moverse (figura 3.54.).

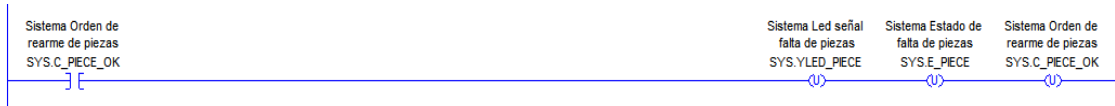


Fig.- 3.54. Activación del tag *F201.C_PIECE_OK*. Fuente: (propia).

Seguidamente la secuencia avanza hasta el otro lado del SFC (*Step_031*) donde se ejecuta un salto a la subrutina de alimentación del rodamiento a la siguiente FAS. Cuando la *Step_036*, último paso de la rutina de alimentación, esté completada, la secuencia salta a la subrutina de medición del rodamiento, ya en la FAS204. Dependiendo de si la medición ha sido correcta o no, se activará un *Step* u otro.

En la siguiente parte se observa las acciones pertinentes dependiendo en cada caso de la altura del rodamiento.

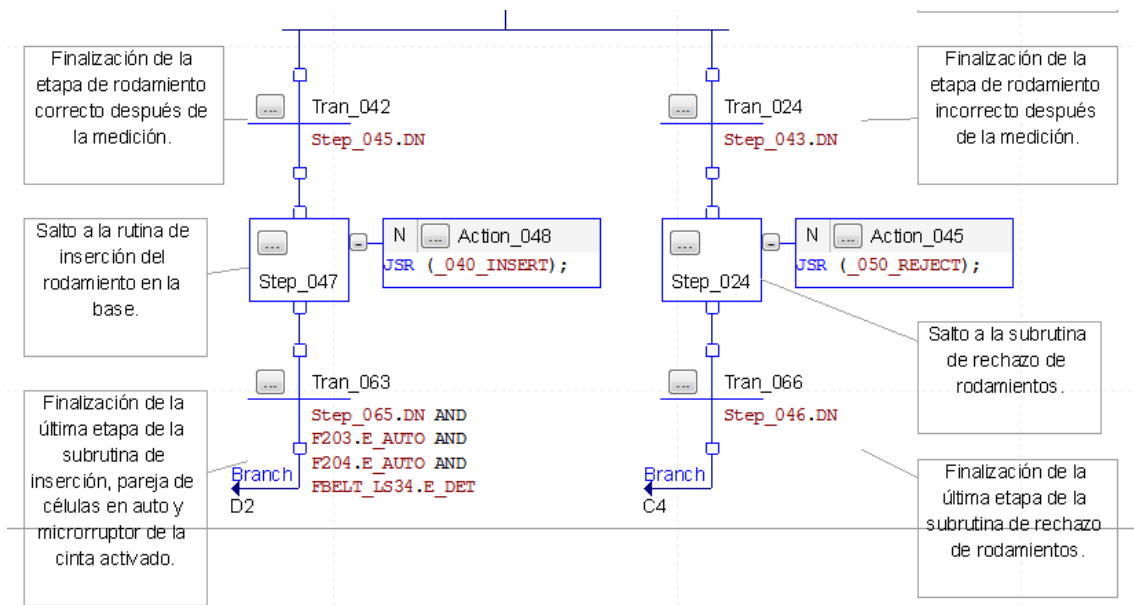


Fig.- 3.55. Parte 3 del SFC de la segunda pareja. Fuente: (propia).

Si la medición del rodamiento se completa de manera correcta, el SFC sigue el camino de la izquierda y procede a saltar a la subrutina de inserción de rodamiento en la base, presente en el *palet* de la cinta. Una vez esto se complete, como se aprecia en la figura 3.55. el sistema espera de nuevo las condiciones iniciales con las que se empieza una nueva secuencia: las dos células en automático y activación del microrruptor de la cinta.

Por el caso contrario, si la subrutina de medición acaba con un resultado negativo, el SFC toma el camino de la derecha y salta a la subrutina de rechazo de rodamiento. Una vez esta subrutina este completada, el programa vuelve a la rutina principal y hace volver al SFC a la primera parte, en el momento que se orden introducir un nuevo rodamiento al sistema, ya que el anterior ha sido rechazado.

FAS207 - FAS208

Igual que sucede en las otras dos parejas, en el interior del SFC principal existen saltos a otras subrutinas de subprocesos de la pareja de células. Estos subprocesos se pueden ver con mejor detalle en el apartado 2.2.

La actual pareja se encarga de alimentar el sistema del último componente que falta, el eje del mecanismo de giro. Por otro lado, gracias a los sensores que posee, puede seleccionar el tipo de material del eje que introducimos al producto final.

Por otro lado, se procede a explicar cuales son las posiciones del plato de la FAS207 y los procesos que se desarrollan en cada una de ellas para así poder comprender mejor el funcionamiento general del sistema. Las posiciones están planteadas para tener un registro que indica su contenido en cada momento de avance del plato.

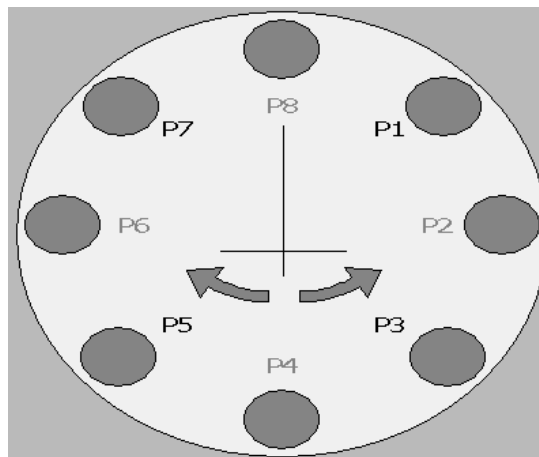


Fig.- 3.56. Representación del plato con todas sus posiciones. Fuentes: (propia).

Como se observa a la figura 3.56. el plato está compuesto por las siguientes 8 posiciones:

- P1: suministros de piezas.
- P2: vacío.
- P3: comprobación de la orientación del eje.
- P4: detección del tipo de material del eje (aluminio o nylon).
- P5: rechazo de ejes incorrectos.
- P6: vacío.
- P7: inserción del eje correcto en la base.
- P8: vacío.

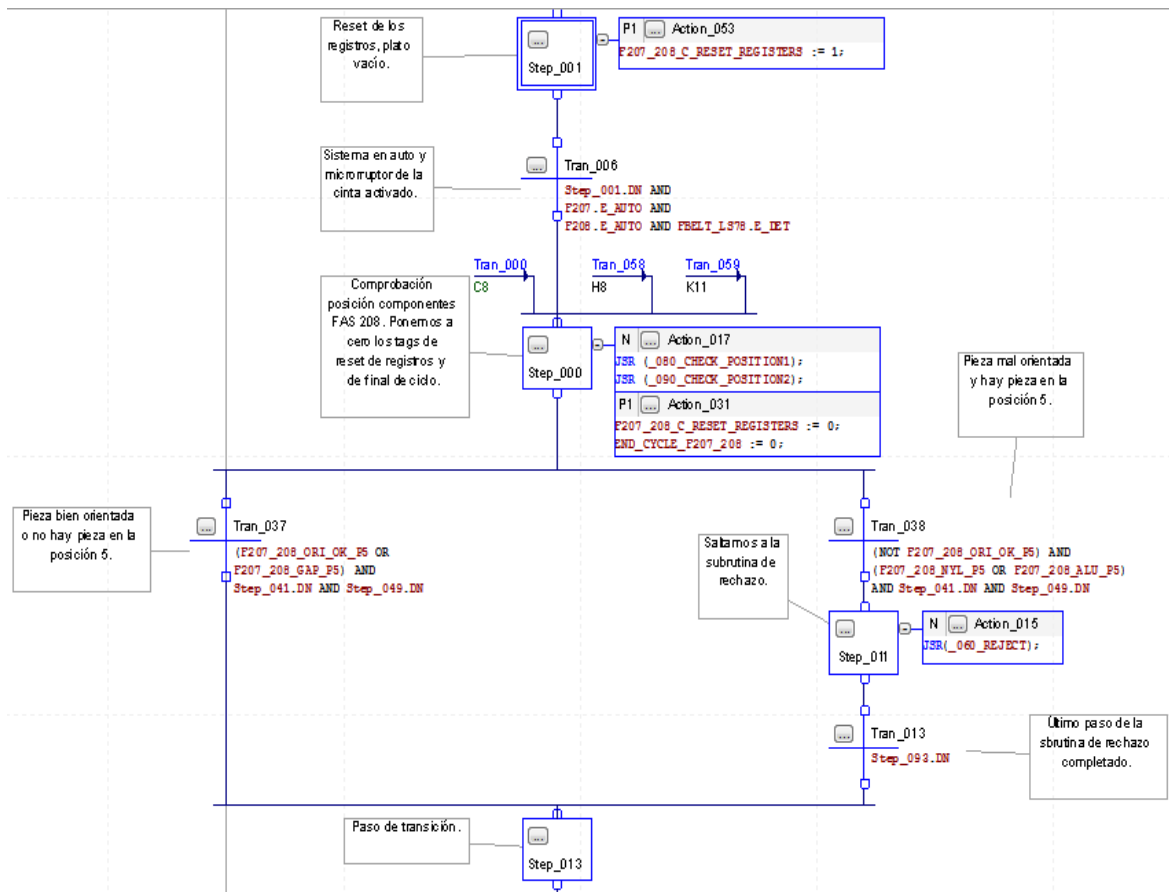


Fig.- 3.57. Primera parte del SFC de la tercera pareja. Fuente: (propia).

- 1) Al empezar la secuencia (figura 3.56.) se activa el *tag* de reseteo de los registros de las posiciones de los platos para que cuando carguemos el programa por primera vez todo este vacío. La primera condición a tener en cuenta es que las dos células se encuentren en automático y que el microrruptor de la cinta este activado. Seguidamente se llevan a cabo 4 acciones, dos de ellas son saltos a subrutinas de comprobación de posición de los actuadores, para evitar que cualquiera de ellos esté mal posicionados al empezar la secuencia. Por otro lado, se resetea la variable de final de ciclo (*END_CYCLE_F207_208*) como se hace también en las otras dos parejas de células.

Seguidamente el SFC llega a dos posibles caminos, el de la izquierda tiene como condiciones que la orientación de la pieza situada en la posición 5 sea correcta o que no haya pieza en dicha posición, entonces pasa al estado de transición siguiente. Si el SFC toma el camino de la derecha, es que la orientación del eje en la posición 5 no es correcta y que el material del eje sea de aluminio o de nylon, mejor dicho, simplemente que haya existencia de eje. Seguidamente y como el eje no tiene la orientación adecuada, se salta a la subrutina de rechazo del eje del sistema. Cuando el *Step_093* este listo, (último de la subrutina de rechazo) la secuencia va a continuar hacia la segunda parte.

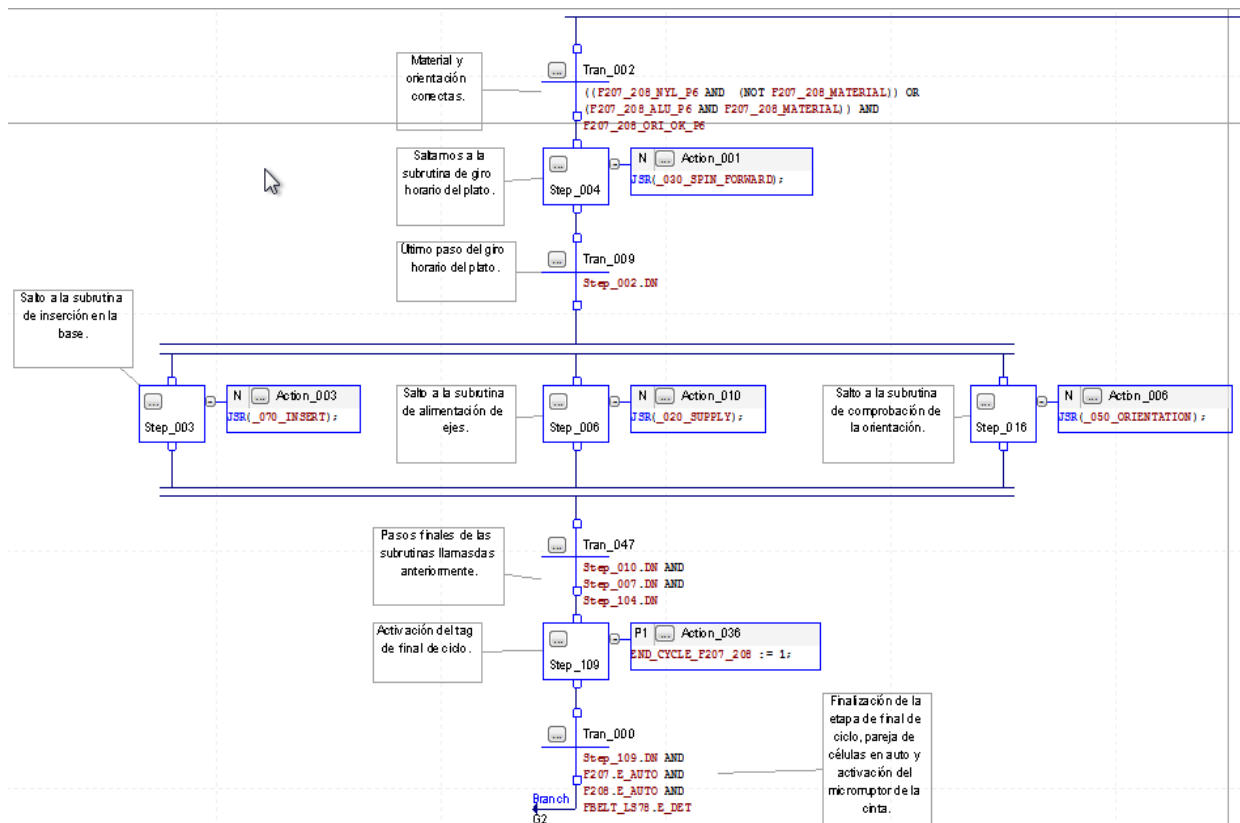


Fig.- 3.58. Segunda parte del SFC de la tercera pareja. Fuente: (propia).

- 2) Una vez pasado el *Step_013* hay tres posibles caminos que tomar (figura 3.57.). En este caso se procede a comentar el camino de la izquierda. Para que la secuencia tome este camino se debe cumplir que el material del eje situado en la posición 6 concuerde con el elegido por el usuario y que la orientación en esa misma posición sea correcta. Si esto se cumple el plato avanza una posición en sentido horario. Una vez hecho esto se producen tres acciones simultáneas: inserción del eje en la base en la P7, alimentación de un nuevo eje en el sistema en la P1 y comprobación de la orientación en la P3. Cuando todos los subprocesos estén listos se activa el *bit* de fin de ciclo (*END_CYCLE_F207_208*) y la secuencia se mantiene a la espera de que se cumplan las condiciones de inicio de nuevo de la secuencia principal: ambas células de la pareja en automático y microinterruptor de la cinta activado. Esto hace volver al SFC al inicio del mismo y empezar una nueva secuencia.

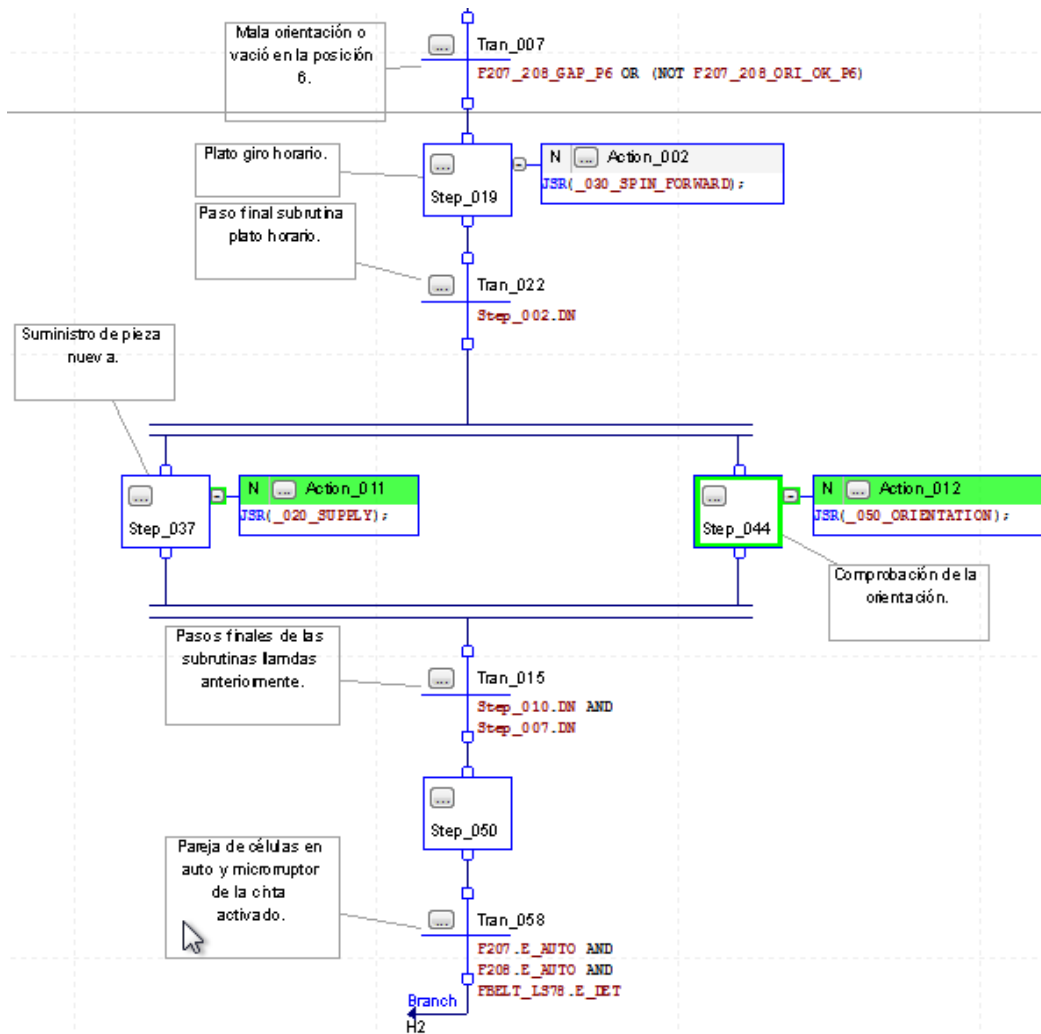


Fig.- 3.59. Tercera parte del SFC de la tercera pareja. Fuente: (propia).

- 3) El camino central se produce si la posición 6 no hay eje o si la orientación no es correcta. Si la orientación no es correcta, en la posición 5 los ejes son rechazados directamente y entonces esa posición queda vacía, pero a ojos del sistema una posición esta vacía solamente cuando el sensor capacitivo no detecta nada. De esta manera cuando se rechaza una pieza mal orientada, su registro llega hasta la posición 6 para que la secuencia tome el camino central, donde se trata la condición de posición 6 vacía, que es lo que ocurre cuando se rechaza un eje en la posición 5. Seguidamente el plato avanza, cuando termina se realizan dos acciones simultáneamente, suministro de nueva pieza al sistema y comprobación de la orientación. Cuando las acciones se completan, la condición para seguir con la secuencia es que las células sigan en automático y el microinterruptor de la cinta este activo. Como se observa, en este caso no se activa ningún *bit* de final de ciclo ya que no se ha insertado ningún eje en la base, por tanto, la secuencia vuelve al inicio para volver a comprobar un nuevo eje, en busca de un eje que cumpla con las condiciones de material y de orientación.

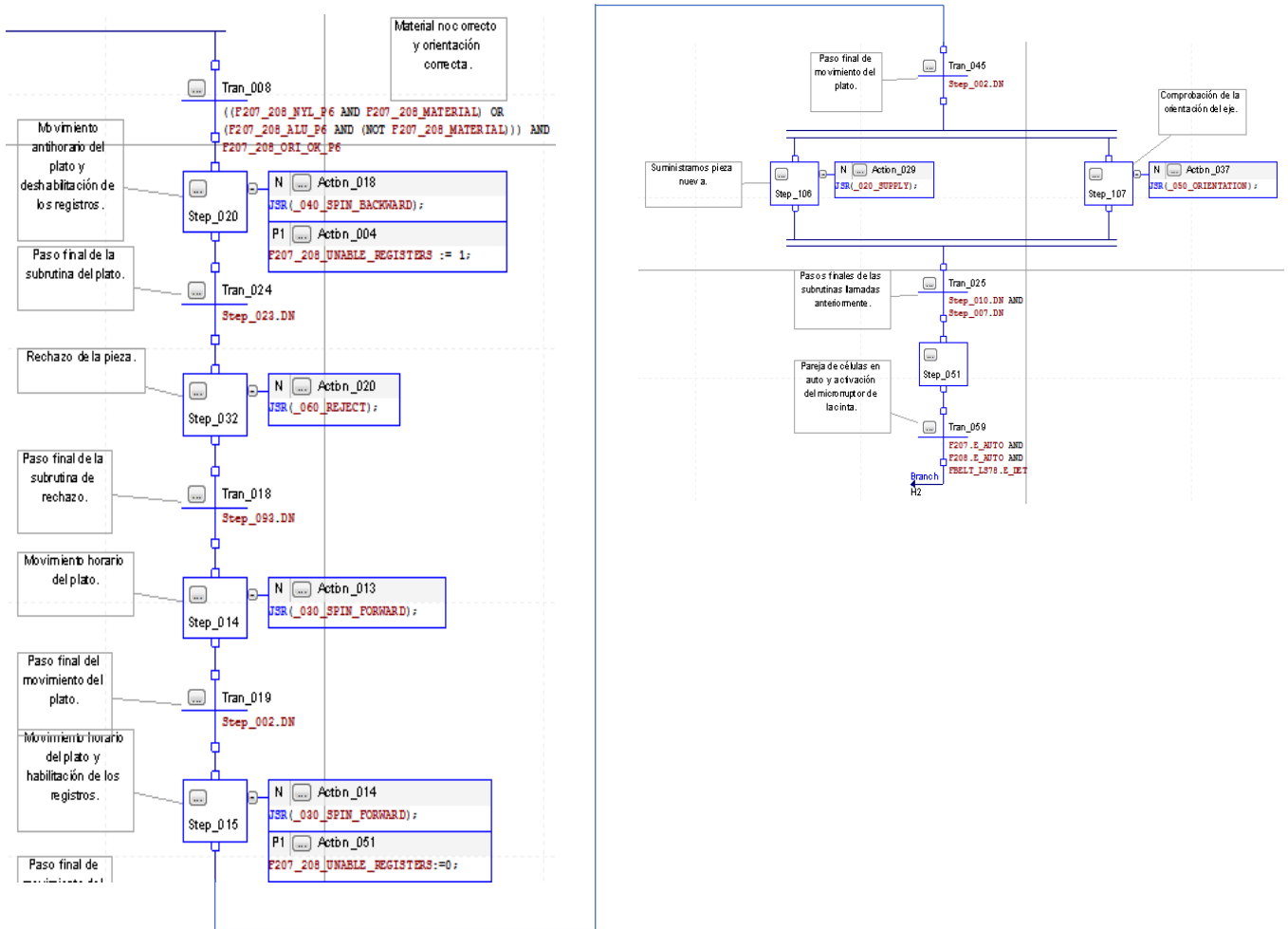


Fig.- 3.60. Cuarta parte del SFC de la tercera pareja. Fuente: (propia).

- 4) Por último, el camino de la derecha se lleva a cabo cuando en la posición 6 el material del eje no coincide con el material que el usuario ha escogido. De esta manera, el plato tira hacia atrás para realizar el proceso de rechazo y una vez completado, gira dos veces hacia delante para recuperar el retroceso y llegar a la nueva posición. Seguidamente se salta a las subrutinas de alimentación de ejes al sistema y de comprobación de la orientación. Una vez acabadas estas subrutinas, el sistema se mantiene a la espera de la condición inicial para poder comenzar de nuevo la secuencia desde el inicio. Al igual que pasa en la tercera parte del SFC (camino central), no se ha insertado ningún eje en la base es por este motivo que no se activa ningún *bit* de final de ciclo, ya que la secuencia se debe de volver a realizar hasta poder insertar un eje correcto en la base.

3.7. Programa de la aplicación SCADA

El presente apartado profundiza en el programa de la aplicación SCADA, creada para poder tener un control remoto de todas las células automatizadas desde el PC. Mediante el software *RSLink* se ha establecido conexión entre el programa del controlador (*RSLogix 5000*) y el programa de SCADA, en este caso el *InTouch Wonderware*.

3.7.1. Árbol de navegación de la aplicación

En la figura 3.61. se aprecia un esquema gráfico de la organización que sigue la aplicación SCADA. A partir de la pantalla inicial (principal) podemos acceder mediante un solo paso a las ventanas de información principal, las ventanas de todas las parejas de FAS, los registros de producción y la pantalla de alarmas del sistema.

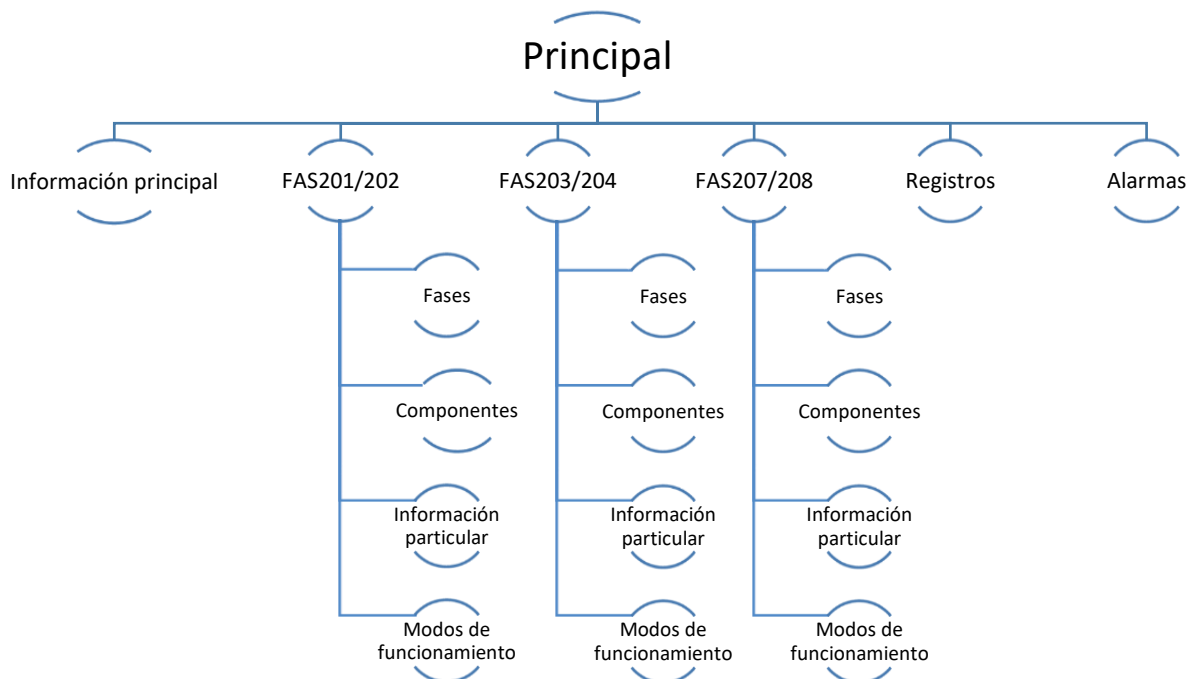


Fig.- 3.61. Árbol de navegación de la aplicación SCADA. Fuente: (propia).

Una vez en la ventana principal de cada pareja de la FAS, como vemos en el árbol de navegación, podemos acceder a las diferentes fases de la secuencia de cada una de ellas, mediante los botones que hay habilitados para dicho efecto. De igual manera se puede acceder a las pantallas de componentes, pulsando sobre los iconos del elemento que queramos visualizar sus controles. También se dispone de un botón de acceso a la pantalla de información, cada pareja cuenta con una donde se hace breve resumen de la funcionalidad y el proceso que se lleva a cabo. Por último, pulsando en el nombre de la pareja de FAS en la que estemos, se tendrá acceso a la ventana de modos de funcionamiento, donde podremos elegir el modo de funcionamiento, automático, manual o mantenimiento, al mismo tiempo se podrá ver si el sistema está en remoto o si el sistema se encuentra en alarma, mediante la misma ventana podremos resetear las alarmas de sistema general.

3.7.2. Definición del tipo de datos y tags

El tipo de datos utilizados en la creación de los *tags* presentes en la aplicación han sido los siguientes:

- I/O Discrete: tipo de datos booleanos, esto quiere decir, con solo dos valores posibles, o bien '0' o bien '1'. Son tipos de datos que no son internos del propio programa, sino que provienen de señales de entrada y salida externos, por este motivo se llaman "I/O".
- I/O Real: tipo de datos que representan valores de numero reales, estos tipos de datos se han utilizado para *tags* que deben representar valor o cantidades de una cierta variable del sistema, como por ejemplo el valor del sensor analógico que mide la altura del rodamiento. Podemos ver este *tag* en la figura 3.62. como F204_AS01.MINS.

Estos tipos de datos son los mismo que los que hay presentes en el programa del controlador, posteriormente en la creación de los mismos tags en el entorno de la aplicación SCADA se han enlazado con los *tags* ya presentes en el programador. De esta manera el valor de estos *tags* se va actualizando en tiempo real mientras el sistema esta en funcionamiento.

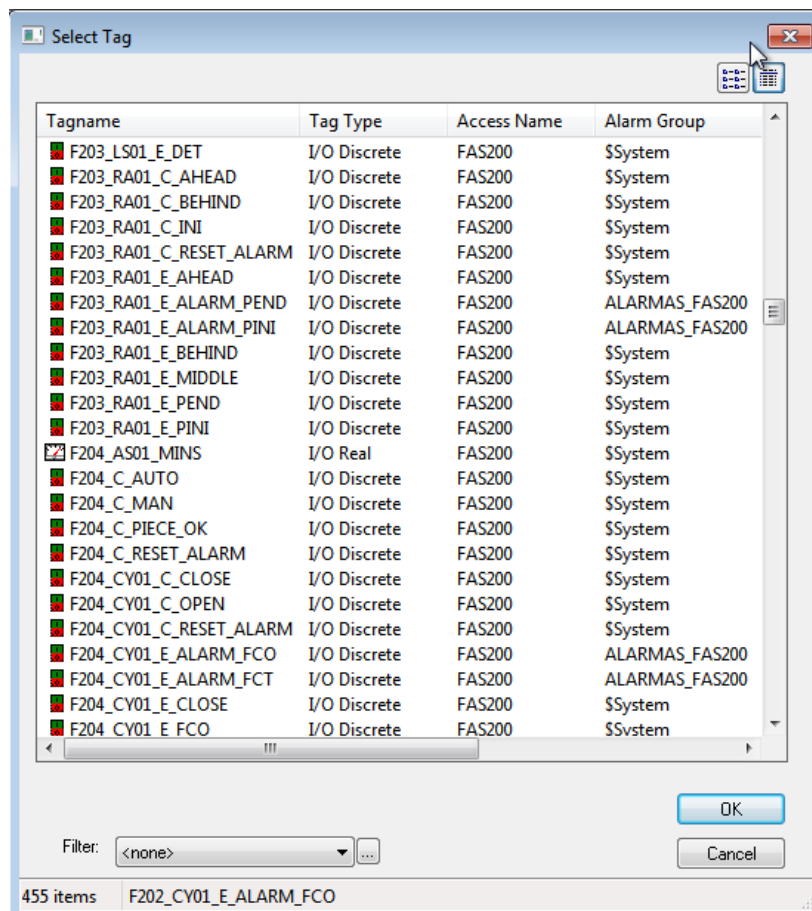


Fig.- 3.62. Tags creados para la aplicación SCADA. Fuente: (propia).

Por otro lado, se ha creado un nuevo grupo de alarmas, independiente del que se crea de forma automática en el momento de crear un nuevo proyecto.

Esto se ha hecho para poder filtrar en el momento que elijamos que alarmas queremos visualizar en la pantalla de alarmas.

Como se puede comprobar en la figura 3.62. solo los *tags* que representan un estado de alarma (...E_ALARM) tiene un *Alarm Group* diferente (ALARMAS_FAS200).

3.7.3. Scripts

Los *scripts* nos permiten programar el funcionamiento del SCADA para que realice acciones cuando nosotros lo deseemos.

En la figura 3.63. podemos ver que se esta utilizando el *script* en el momento del arranque del *Run Time* de la aplicación, esto lo podemos ver en la parte superior donde pone: *On Startup*, en este caso programamos que ventanas queremos que se abran en el momento de inicializar la aplicación y establecer la comunicación con el controlador.

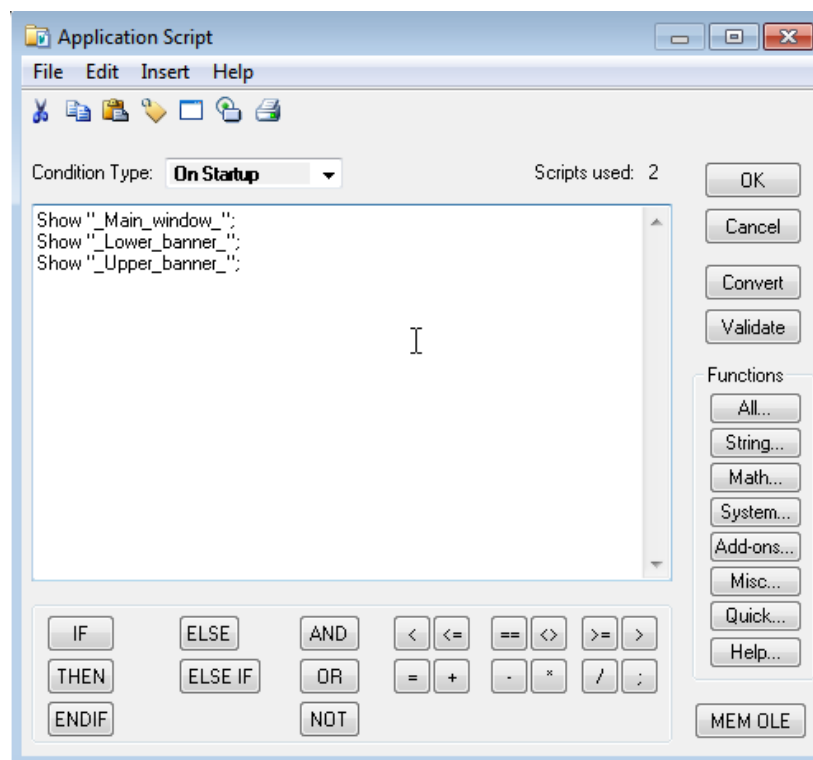


Fig.- 3.63. Script On Startup de la aplicación SCADA. Fuente: (propia).

Por otro lado, en la figura 3.64. se puede apreciar como tenemos otro *Script*, pero en este caso no es del mismo tipo, ya que si nos fijamos en el *Condition Type* pone: *While Running*. Esto significa que el *script* se va a estar repitiendo todo el rato mientras el programa este activo, ejecutándose cada 50 ms, este es un parámetro que se puede variar a medida del programador.

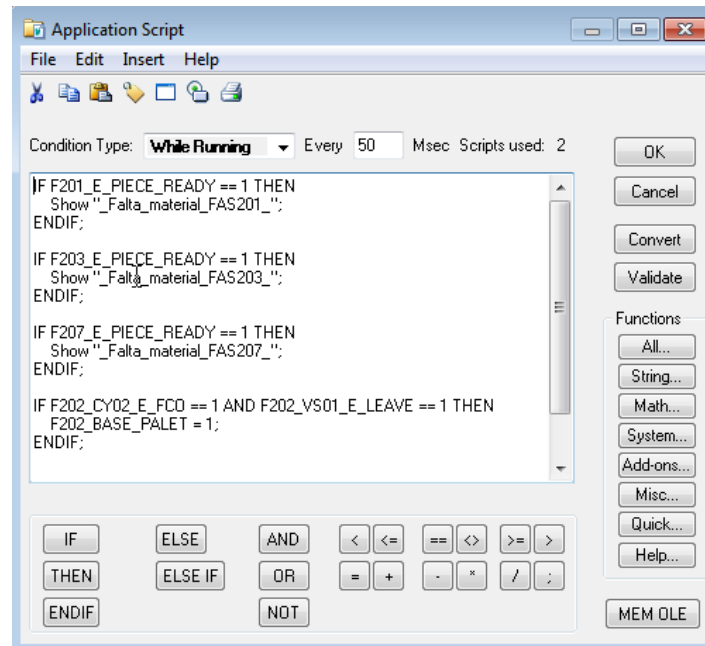


Fig.- 3.64. Script While Running de la aplicación SCADA. Fuente: (propia).

Este segundo *script* se ha utilizado para trabajar con variable internas del programa que nos sirven para poder hacer aparecer/desaparecer ventanas como vemos en el ejemplo o para cambiar de estado, color, apariencia cuando una condición se cumpla.

3.7.4. Diseño de las ventanas principales de la aplicación

En la figura 3.65. se pueden ver todas las ventanas del sistema, en este apartado se comentan las principales, en los siguientes apartados se tratan las restantes.

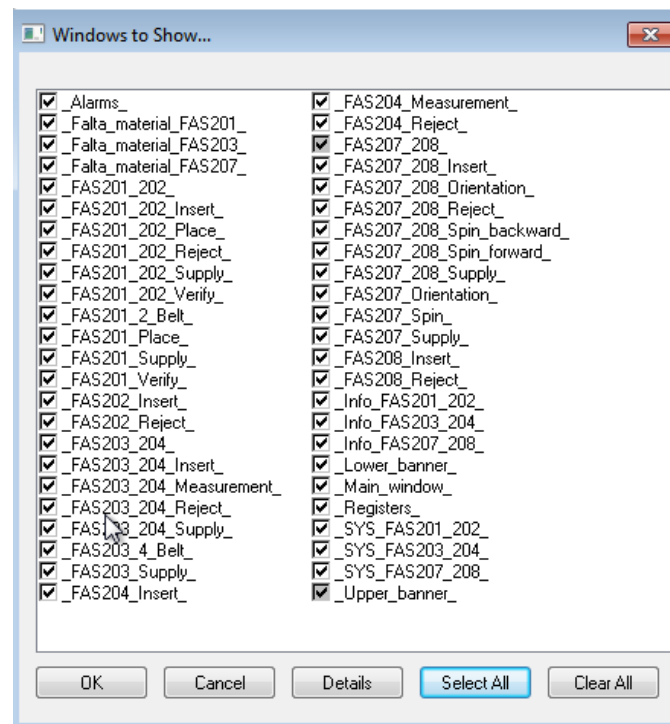


Fig.- 3.65. Ventanas de la aplicación. Fuente: (propia).

El programa consta de ventanas que se van abriendo y cerrando en función de los botones que pulsemos, pero hay dos de ellas que nunca se cierran, estas son los *banners* inferior y superior.

En la figura 3.66. se puede apreciar el *banner* superior, por otro lado, en la figura 3.67. se observa el *banner* inferior.



Fig.- 3.66. Banner superior de la aplicación. Fuente: (propia).

Mediante el *banner* superior podemos acceder a las pantallas de todas las parejas de FAS y visualizar su sinóptico, a la información del sistema, a la ventana de registros de producción y a la pantalla de alarmas.

Fig.- 3.67. Banner inferior de la aplicación. Fuente: (propia).

El *banner* inferior sirve meramente de información para el usuario, no posee ningún tipo de acceso a ninguna ventana, nos informa del nombre del proyecto y el autor.

Siguiendo con el análisis, en la figura 3.68. se encuentra la ventana principal del sistema, al igual que el *banner* inferior, sirve solo de información de las maquinas que forman el sistema, se puede acceder a ella mediante el botón de información del *banner* superior.



Fig.- 3.68. Ventana principal de la aplicación. Fuente: (propia).

Al igual que en el sistema, las ventanas de las células se han diseñado por parejas. En las figuras 3.69., 3.70., 3.71. se pueden observar las ventanas de las parejas de células, que se encuentran comentadas detalladamente en la tabla 3.16.

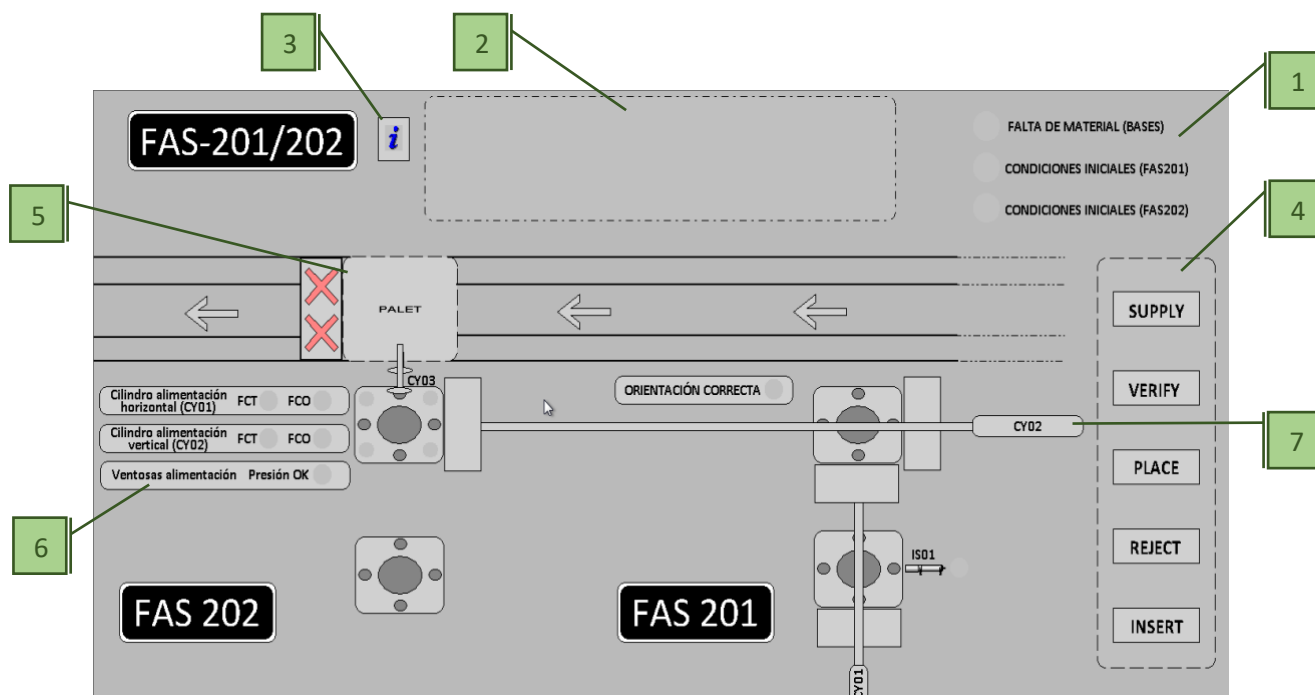


Fig.- 3.69. Pantalla principal de la pareja FAS-201/202. Fuente: (propia).

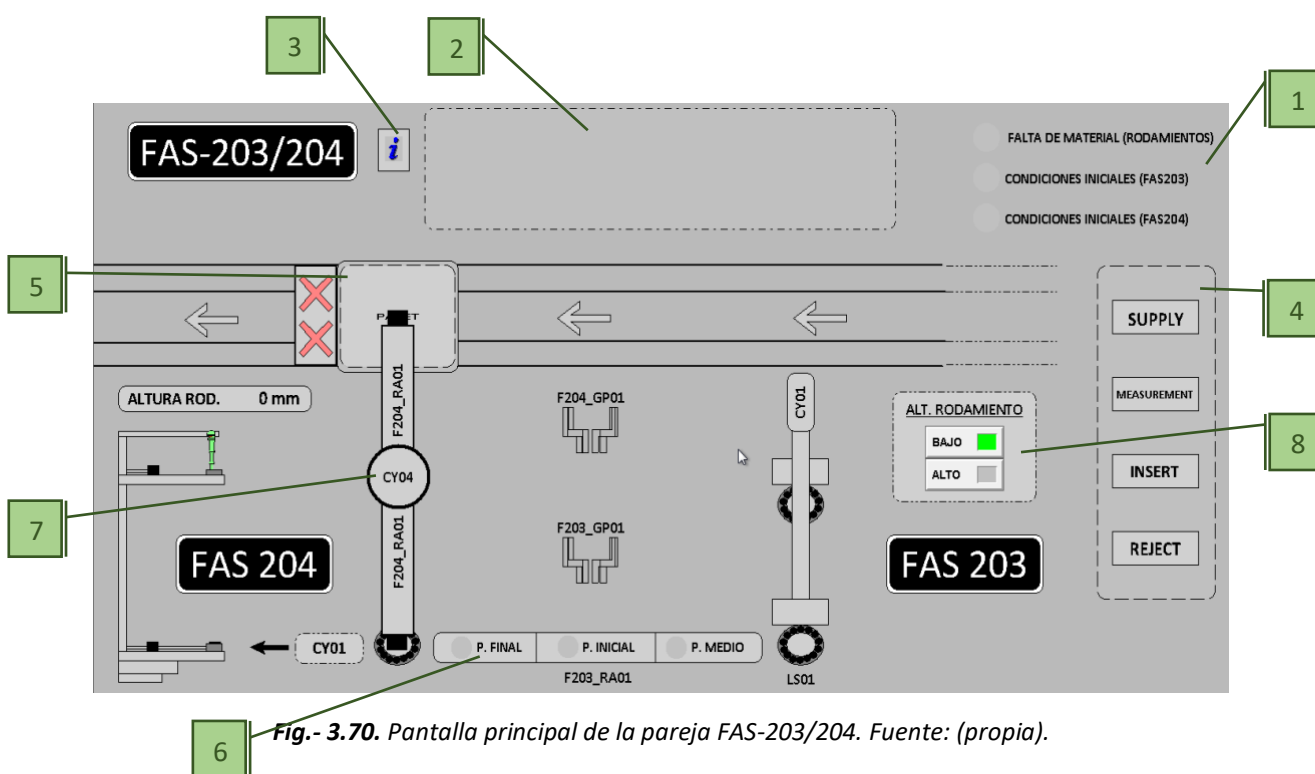


Fig.- 3.70. Pantalla principal de la pareja FAS-203/204. Fuente: (propia).

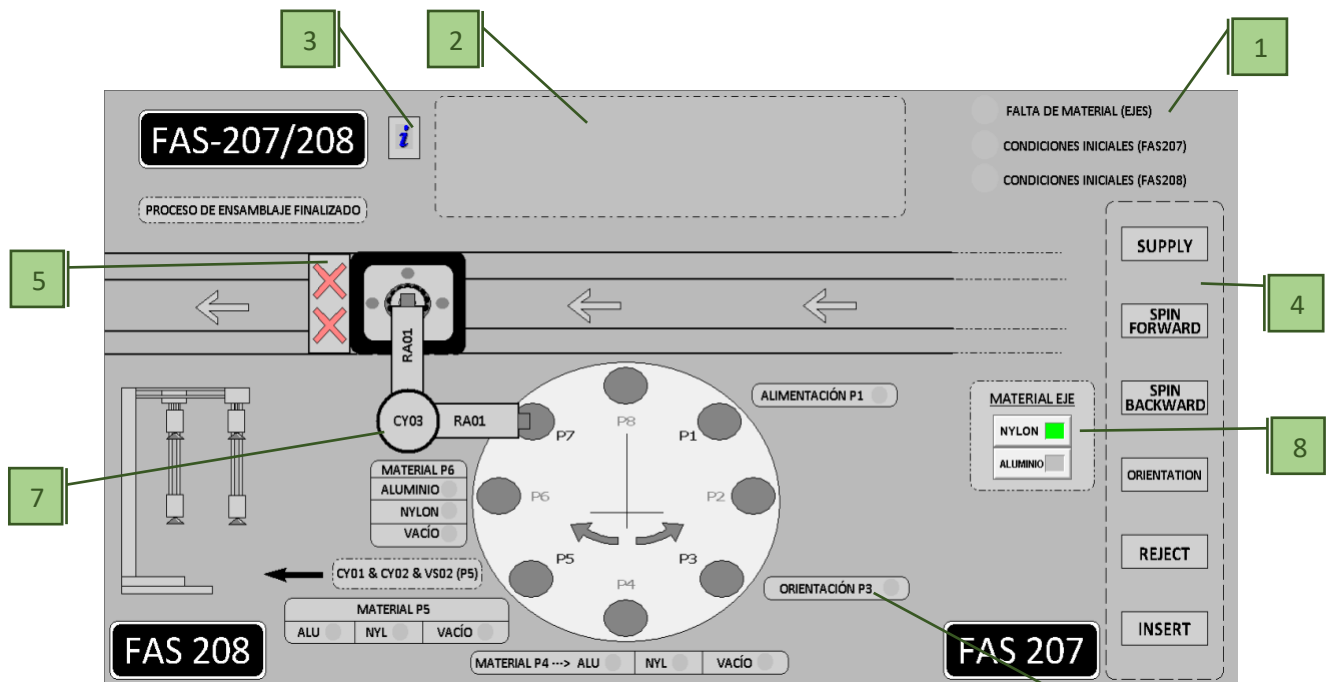


Fig.- 3.71. Pantalla principal de la pareja FAS-207/208. Fuente: (propia).

Se han adjuntados numero a todas las pantallas principales indicando los aspectos más importantes, todos los numero son iguales en las tres pantallas, menos el número 8, que solo es común para las dos últimas parejas. Por este motivo, la tabla 3.16. es común para la explicación de las tres parejas.

Número	Descripción
1	Indicadores comunes a las tres parejas de células, nos indican la falta de material, bases en el caso de la FAS201, rodamientos en el caso de la FAS203 o ejes en el caso de la FAS207. Por otro lado, nos indican si las células se encuentran en ejecución de su etapa de condiciones iniciales.
2	Este espacio está reservado para la pantalla de sistema donde se puede observar y manipular el modo de funcionamiento del sistema, también se puede resetear la alarma general del sistema.
3	Este botón sirve para acceder a la pantalla de información, cada pareja posee una diferente, en ella podemos observar una breve explicación de la funcionalidad de las células.
4	En este espacio se encuentran los botones que dan acceso a las pantallas de proceso, cada pareja tiene la suya. Cada botón da acceso a un proceso/fase diferente en la que podemos ver el SFC/GRAFSET representando el proceso y su evolución.
5	Acceso directo a la pantalla de componente del cilindro de la cinta, de esta manera se puede manipular el cilindro directamente. En el de la pareja compuesta por la FAS203 y la FAS204, aparecen dos cilindros, el de los <i>stoppers</i> y el cilindro elevador del <i>palet</i> .

6	Nos permite el acceso directo a la pantalla de componente. En esta pantalla podemos visualizar el estado de los sensores del actuador y dar ordenes de movimiento al actuador.
7	Acceso directo a la pantalla de componente, esto tiene la misma función que el número 6. Se diferencian en que los que son como el número 6 debemos pulsar en un recuadro de texto donde nos sale el nombre del componente al que vamos a acceder. Por otro lado, los actuadores como el número 7, posee un dibujo creado en el SCADA. Se diferencia en la forma de acceso a ellos, pero la pantalla a la que se accede tiene las mismas funcionalidades, dependiendo del actuador al que se acceda.
8	Elección del tipo de producto final que se quiera obtener. En la segunda pareja, debemos escoger la altura del rodamiento, entre bajo o alto, en la tercera pareja se debe escoger el material del eje, nylon o aluminio. En la primera pareja no aparece esta posibilidad, ya que no posee ningún tipo de filtro de material o fisonomía para las bases.

Tabla 3.16. Puntos importantes del diseño de las ventanas principales. Fuente: (propia).

En la figura 3.72. encontramos la pantalla de modo de funcionamiento de cada pareja de células. Esta pantalla es conjunta de ambas maquina de la pareja que forman. En la tabla 3.17. se encuentra detallada los puntos más importantes.

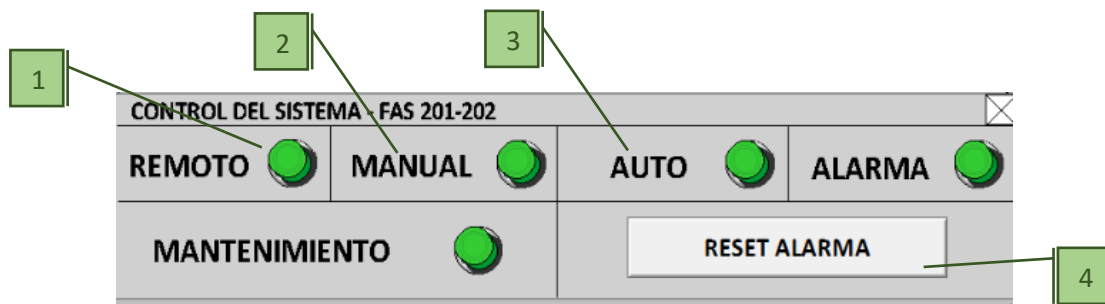


Fig.- 3.72. Pantalla de modo de funcionamiento del sistema. Fuente: (propia).

Número	Descripción
1	Todos los señalizadores luminosos se iluminan en el momento que el estado que indican este activo, en el caso de la alarma se ilumina en rojo, cuando se encuentra en remoto se ilumina en azul igual que la FAS, los demás se iluminan en verde.
2	Este texto, además de mostrar que el indicador luminoso muestra el estado de manual, también sirve para pasar las dos células, al mismo tiempo, a manual.
3	Al igual que el texto de manual, tiene dos funciones indicador de texto y también pasa a automático, al mismo tiempo, a las dos células.
4	Botón de <i>reset</i> de las alarmas generales de las dos células que forman la pareja en cuestión.

Tabla 3.17. Puntos importantes del diseño de la ventana de modo de funcionamiento. Fuente: (propia).

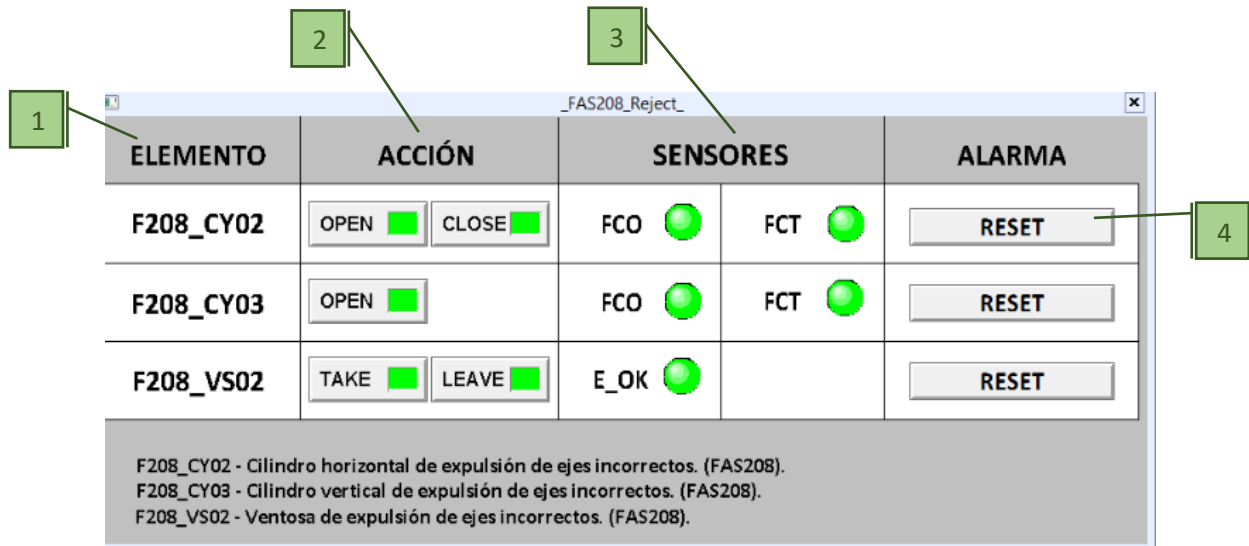


Fig.- 3.73. Pantalla de descripción y manipulación de componente. Fuente: (propia).

Número	Descripción
1	Nos indica el componente que se va a manipular y del que estamos viendo el estado de sus sensores.
2	Muestra los estados del componente y mediante los botones somos capaces de dar órdenes para cambiar el estado de los actuadores.
3	Podemos apreciar el estado de los sensores del actuador y así saber la posición en la que se encuentra sin tener que tomar contacto visual con dicho actuador.
4	Botón de <i>reset</i> de la alarma del componente, en el momento que la alarma esta activa, el botón se encuentra en constante parpadeo de color rojo, alertando del estado de alarma del componente.

Tabla 3.18. Puntos importantes de la pantalla de componentes del sistema. Fuente: (propia).

Como se observa en la figura 3.73. se han agrupado por pantallas los actuadores/componentes que tengan un subproceso en común. En el ejemplo que se ha explicado, vemos que es el subproceso de rechazo de ejes de la FAS208, en la que intervienen dos cilindros (F208_CY02 y F208_CY03) y por otro lado la ventosa que transporta el objeto de posición (F208_VS02). Una vez en la ventana principal de alguna de las parejas de FAS, se puede acceder a estas pantallas mediante los componentes del sistema. Alguna pantalla se podrá acceder desde componentes distintos, esto se produce debido a que comparten el proceso al que se accede, al igual que en el ejemplo anterior.

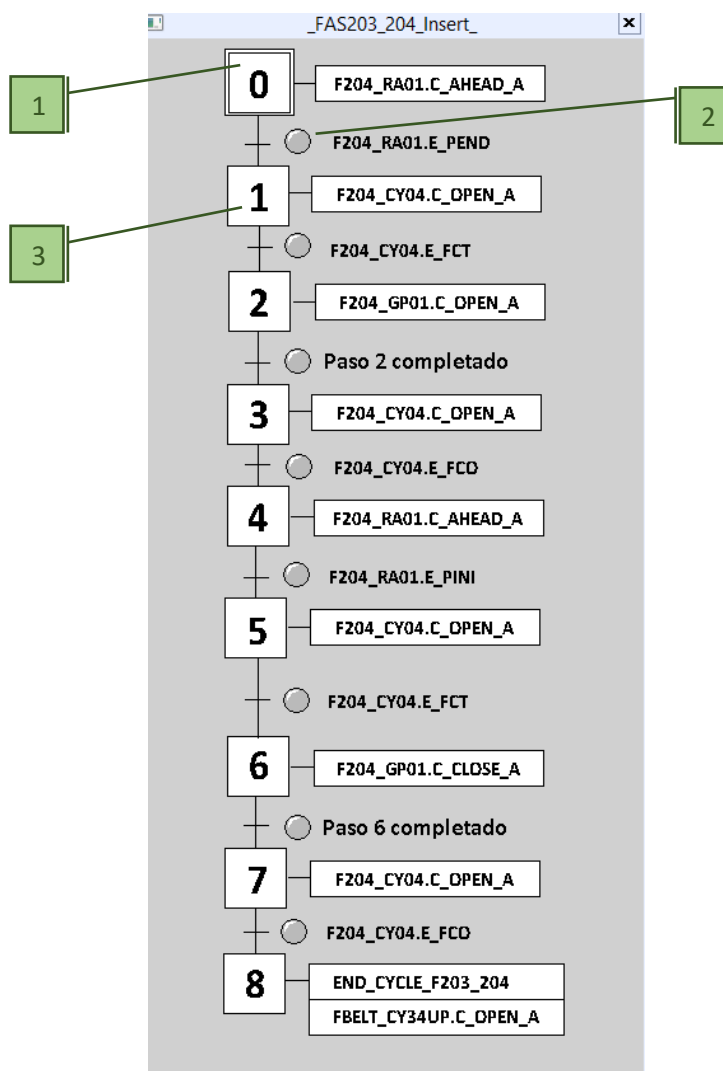


Fig.- 3.74. Proceso de inserción de rodamiento en la base. Fuente: (propia).

Número	Descripción
1	Acción principal del SFC, se distingue por su doble cuadrado. Es la primera acción que se realiza en el instante que se inicia la secuencia.
2	Condición para que la secuencia siga avanzando, las transiciones son condiciones que el sistema debe de cumplir para que la secuencia siga su curso.
3	Acción normal de la secuencia se activa en el momento que su transición (condición) previa se activa y deja seguir a la secuencia. Una vez se activa, la secuencia sigue y espera a cumplir la siguiente condición/transición.

Tabla 3.18. Puntos importantes de la ventana de secuencia. Fuente: (propia).

3.7.5. Diseño de la ventana de alarmas de la aplicación

El sistema posee alarmas en casi todos los componentes con los que cuenta, más las alarmas de sistema que se ven activadas en el momento que la alarma de algún componente este activa. Esto se produce debido a que disponemos de sensores que tienen la capacidad de plasmarnos mediante señales eléctricas lo que sucede en nuestro entorno. Mediante estas señales eléctricas podemos crear alarmas para las condiciones que creamos oportunas catalogarlas como de emergencia.

Debido a esto, en la aplicación SCADA se cuenta con una ventana de alarmas, figura 3.75., que en todo momento notifica de las alarmas activas y las no activas y nos notifica si están reconocidas o no.

[illegible]

Fig.- 3.75. Ventana de alarmas del sistema. Fuente: (propia).

Se observa que la pantalla de alarmas cuenta con las siguientes columnas:

- Date: fecha en la que se ha producido la alarma.
- Time: hora en la que se ha producido la alarma.
- State: este puede ser *UNACK* (no reconocida) o bien *ACK* (reconocida).
- Name: nombre de la alarma.
- Provider: es el que ha preparado todo el programa, normalmente suele aparecer la palabra *InTouch*.
- Comment: Son los comentarios de la alarma activa. En el momento que se reconocen alarmas, poder tomar la decisión de poner un nuevo comentario una vez la alarma este reconocida.

Se aprecia en la esquina superior derecha que hay un botón de reconocimiento de alarmas, con este botón podemos reconocer las alarmas que deseemos, tanto individualmente, como en grupos de más alarmas.

3.7.6. Ventana de registros del sistema

Con el objetivo de mantener un seguimiento cualitativo sobre el sistema, se ha tomada la decisión de incorporar una ventana de registros al sistema, debido a que los datos tratados siempre se muestran en forma de enteros. En la figura 3.76. se observa los registros de cada pareja de FAS, en la primera las bases, en la segunda los rodamientos y en la tercera los ejes, cada una con sus posibles variaciones. También cuentan con un botón de *reset* de los registros, al pulsar los respectivos botones, los registros de la FAS que escojamos se van a reiniciar a cero. Es eficiente mantener un registro de todos los componentes que se van introduciendo al sistema para poder almacenar datos y posteriormente crear BBDD para darle un uso más extenso a algo tan importante como es la información del proceso.



Fig.- 3.76. Pantalla de registros del sistema. Fuente: (propia).

3.7.7. Ventanas secundarias del sistema

A parte de las ventanas ya comentadas, en el sistema existen otras con menos peso, llamadas secundarias. Este grupo cuenta con las ventanas de información de las tres parejas y con las ventanas de falta de material.

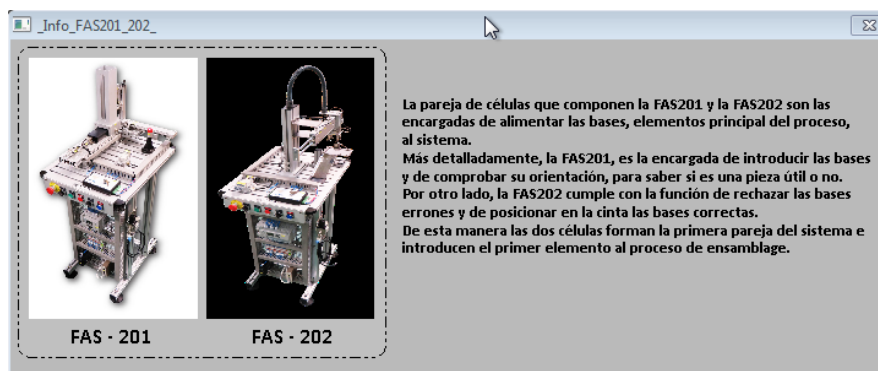


Fig.- 3.77. Ventana de información de las parejas de células. Fuente: (propia).

Por 3ltimo, contamos con la ventana de falta de piezas, que como podemos ver en el *script* de la figura 3.64., siempre aparecer3 cuando se active el *SYS_E_PIECE*. En la figura 3.78. observamos el mensaje de alerta del sistema, aparece un bot3n que debemos pulsar en el momento que ya hayamos alimentado al sistema de bases, rodamientos o ejes, dependiendo de donde proceda la se1al. Dependiendo la se1al, siempre se activar3 una ventana muy parecida solo cambiando el nombre del material que falta.

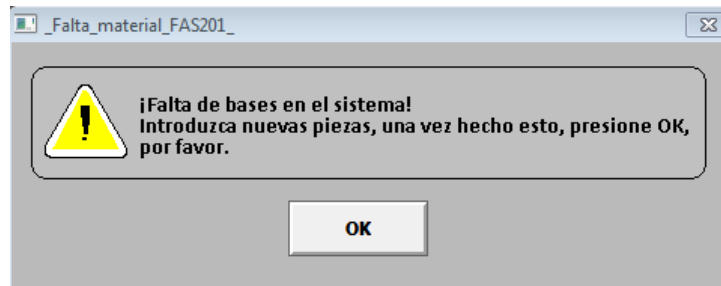


Fig.- 3.78. Falta de piezas en el sistema. Fuente: (propia).

4. Pruebas y resultados

En este apartado se procede a explicar y exponer las pruebas realizadas al sistema y los resultados obtenidos.

4.1. Pruebas

En cuanto a las pruebas se refiere, hay dos tipos de pruebas a tener en cuenta, estas son las FAT (*Factory Acceptance Test*) las cuales representan aquellos tests que se han realizado antes de probar directamente la programación sobre la planta real. Por otro lado, tenemos las SAT (*Site Acceptance Test*), son aquellas pruebas que se realizan *in situ*, en nuestro caso, en el laboratorio, volcando la programación sobre la FAS200.

Para poder realizar las FAT y para poder comprobar la conectividad entre el programa del controlador y la aplicación SCADA se ha utilizado el *software RSLogix Emulate 5000 Chassis*. Este programa nos ayuda a poder crear un PLC emulado con el que podemos establecer conexión. Por otro lado, las pruebas FAT de funcionalidad se han llevado a cabo con el mismo *software* y forzando valores en las entradas del PLC emulado para poder ver si el comportamiento del controlador era el adecuado.

Las pruebas SAT se han llevado a cabo íntegramente en el laboratorio de automática. Primero se obtuvo conexión entre el PLC principal y el programa del controlador, seguidamente entre el controlador principal (*CompactLogix*) y los controladores independientes de cada FAS (*MicroLogix*). Una vez lograda esta conexión se logró conectar el programa del controlador con la aplicación SCADA.

Una vez se obtuvo conexión, mediante la activación de los sensores de planta, se veían reflejados su estado en el SCADA. Por otro lado, se ha ido realizando las pruebas sobre los sensores y actuadores, de cada célula como podemos ver en las tablas presentes en el siguiente apartado.

4.2. Resultados

A continuación, se presentan todas las pruebas realizadas al sistema y a todos los componentes que lo forman.

Debemos destacar que todas las pruebas realizadas con el emulador (pruebas FAT) han sido satisfactorias y rápidamente se pudo acceder al laboratorio a realizar las SAT.

En cuando a las SAT, los resultados han sido muy satisfactorios, menos en un caso en el que no se ha podido obtener el resultado deseado.

El caso negativo se ha producido en el detector de posición inicial del actuador de giro de la FAS208. No se detecta ninguna señal en el momento en el que el actuador se encuentra en dicha posición. La acción de llegar a la posición inicial se realiza, pero la detección de este estado del actuador rotativo no se genera y por lo tanto no se recibe en el programa del controlador.

Cabe destacar que dicha señal tampoco se ve reflejada en la pantalla LCD de I/O del *MicroLogix*) lo que hace entender que el sensor no está operativo.

A continuación, se muestran los resultados obtenidos de la FAS201:

Elemento	Número	Prueba	Resultado esperado	Valoración
F201	1	Cuando la célula entra en remoto entra al estado por defecto	Modo manual	CUMPLE
	2	Intento de cambiar de modo mientras el sistema esta en alarma	Se puede cambiar de modo en circunstancias de alarma.	CUMPLE
	3	Modo en el que el sistema deja de estar en remoto	No hay ningún modo en el que deje de estar en remoto.	CUMPLE
	4	Activación alarma de un elemento	Activación alarma del sistema	CUMPLE
	5	Activación <i>reset</i> del sistema	Se activa el reseteo de la alarma del sistema siempre y cuando no haya ninguna alarma de componente activa	CUMPLE
	6	Sistema en alarma	No funciona la activación de ningún elemento.	CUMPLE
	7	Sistema en alarma y modo de mantenimiento activado	Es posible poder iniciar condiciones iniciales y activar actuadores	CUMPLE
F201_CY01	1	Orden de abrir (cuando no esta abierto)	Estado de apertura ON	CUMPLE
	2	Orden de abrir (cuando esta abierto)	Estado de apertura OFF	CUMPLE
	3	Detector final de carrera abierto	Estado FCO ON	CUMPLE
	4	Detector final de carrera cerrado	Estado FCT ON	CUMPLE
	5	Estado de apertura activo durante 5s sin detección de FCO	Alarma FCO ON	CUMPLE
	6	Estado de apertura OFF durante más de 5s segundo y FCT inactivo	Alarma FCT ON	CUMPLE
	7	Alarma Elemento ON	Elementos bloqueados	CUMPLE

	8	Orden <i>reset</i> alarma del componente	Alarma se resetea siempre y cuando el estado que las acciona ya no este activo	CUMPLE
F201_CY02	1	Orden de abrir (cuando no esta abierto)	Estado de apertura ON	CUMPLE
	2	Orden de abrir (cuando esta abierto)	Estado de apertura OFF	CUMPLE
	3	Detector final de carrera abierto	Estado FCO ON	CUMPLE
	4	Estado de apertura activo durante 5s sin detección de FCO	Alarma FCO ON	CUMPLE
	5	Alarma Elemento ON	Elementos bloqueados	CUMPLE
	6	Orden <i>reset</i> alarma del componente	Alarma se resetea siempre y cuando el estado que las acciona ya no este activo	CUMPLE
F201_CY03	1	Orden de abrir (cuando no esta abierto)	Estado de apertura ON	CUMPLE
	2	Orden de abrir (cuando esta abierto)	Estado de apertura OFF	CUMPLE
	3	Detector final de carrera abierto	Estado FCO ON	CUMPLE
	4	Estado de apertura activo durante 5s sin detección de FCO	Alarma FCO ON	CUMPLE
	5	Alarma Elemento ON	Elementos bloqueados	CUMPLE
	6	Orden <i>reset</i> alarma del componente	Alarma se resetea siempre y cuando el estado que las acciona ya no este activo	CUMPLE
F201_IS01	1	Detección pieza metálica	Estado detección ON	CUMPLE

Tabla 3.20. Pruebas realizadas a la FAS201.

A continuación, se muestran los resultados obtenidos de la FAS202:

Elemento	Número	Prueba	Resultado esperado	Valoración
F202	1	Cuando la célula entra en remoto entra al estado por defecto	Modo manual	CUMPLE

	2	Intento de cambiar de modo mientras el sistema esta en alarma	Se puede cambiar de modo en circunstancias de alarma.	CUMPLE
	3	Modo en el que el sistema deja de estar en remoto	No hay ningún modo en el que deje de estar en remoto.	CUMPLE
	4	Activación alarma de un elemento	Activación alarma del sistema	CUMPLE
	5	Activación <i>reset</i> del sistema	Se activa el reseteo de la alarma del sistema siempre y cuando no haya ninguna alarma de componente activa	CUMPLE
	6	Sistema en alarma	No funciona la activación de ningún elemento.	CUMPLE
	7	Sistema en alarma y modo de mantenimiento activado	Es posible poder iniciar condiciones iniciales y activar actuadores	CUMPLE
F202_CY01	1	Orden de abrir (cuando no esta abierto)	Estado de abrir ON	CUMPLE
	2	Orden de abrir (cuando esta abierto)	Ninguna acción	CUMPLE
	3	Orden de cerrar (cuando no esta cerrado)	Estado de cerrar ON	CUMPLE
	4	Orden de cerrar (cuando esta cerrado)	Ninguna acción	CUMPLE
	5	Detector final de carrera abierto	Estado FCO ON	CUMPLE
	6	Detector final de carrera cerrado	Estado FCT ON	CUMPLE
	7	Estado de apertura activo durante 5s sin detección de FCO	Alarma FCO ON	CUMPLE
	8	Estado de apertura OFF durante más de 5s segundo y FCT inactivo	Alarma FCT ON	CUMPLE
	9	Alarma Elemento ON	Elementos bloqueados	CUMPLE

	10	Orden <i>reset</i> alarma del componente	Alarma se resetea siempre y cuando el estado que las acciona ya no este activo	CUMPLE
F202_CY02	1	Orden de abrir (cuando no esta abierto)	Estado de apertura ON	CUMPLE
	2	Orden de abrir (cuando esta abierto)	Estado de apertura OFF	CUMPLE
	3	Detector final de carrera abierto	Estado FCO ON	CUMPLE
	4	Detector final de carrera cerrado	Estado FCT ON	CUMPLE
	5	Estado de apertura activo durante 5s sin detección de FCO	Alarma FCO ON	CUMPLE
	6	Estado de apertura OFF durante más de 5s segundo y FCT inactivo	Alarma FCT ON	CUMPLE
	7	Alarma Elemento ON	Elementos bloqueados	CUMPLE
	8	Orden <i>reset</i> alarma del componente	Alarma se resetea siempre y cuando el estado que las acciona ya no este activo	CUMPLE
F202_CY03	1	Orden de abrir (cuando no esta abierto)	Estado de apertura ON	CUMPLE
	2	Orden de abrir (cuando esta abierto)	Estado de apertura OFF	CUMPLE
F202_VS01	1	Orden de coger (cuando no esta cogiendo)	Estado de coger ON	CUMPLE
	2	Orden de coger (cuando esta cogiendo)	Ninguna acción	CUMPLE
	3	Orden de dejar (cuando no lo esta haciendo)	Estado de dejar ON	CUMPLE
	4	Orden de dejar (cuando no lo esta haciendo)	Ninguna acción	CUMPLE
	5	Detección de vacío	Estado OK ON	CUMPLE
	6	Estado de coger ON durante 5s y no se recibe señal del presostato	Alarma de componente ON	CUMPLE
	7	Alarma elemento ON	Elementos bloqueados	CUMPLE
	8	Orden <i>reset</i> Alarma elemento	Alarma elemento OFF	CUMPLE
FBELT_LS12	1	Detección pieza metálica	Estado detección ON	CUMPLE

FBELT_CY12	1	Orden de abrir (cuando no esta abierto)	Estado de apertura ON	CUMPLE
	2	Orden de abrir (cuando esta abierto)	Estado de apertura OFF	CUMPLE

Tabla 3.21. Pruebas realizadas a la FAS202.

A continuación, se muestran los resultados obtenidos de la FAS203:

Elemento	Número	Prueba	Resultado esperado	Valoración
F203	1	Cuando la célula entra en remoto entra al estado por defecto	Modo manual	CUMPLE
	2	Intento de cambiar de modo mientras el sistema esta en alarma	Se puede cambiar de modo en circunstancias de alarma.	CUMPLE
	3	Modo en el que el sistema deja de estar en remoto	No hay ningún modo en el que deje de estar en remoto.	CUMPLE
	4	Activación alarma de un elemento	Activación alarma del sistema	CUMPLE
	5	Activación <i>reset</i> del sistema	Se activa el reseteo de la alarma del sistema siempre y cuando no haya ninguna alarma de componente activa	CUMPLE
	6	Sistema en alarma	No funciona la activación de ningún elemento.	CUMPLE
	7	Sistema en alarma y modo de mantenimiento activado	Es posible poder iniciar condiciones iniciales y activar actuadores	CUMPLE
F203_RA01	1	Orden adelante (cuando no tiene la orden dada)	Estado adelante ON	CUMPLE
	2	Orden adelante (cuando tiene la orden dada)	Ninguna acción	CUMPLE
	3	Orden atrás (cuando no tiene la orden dada)	Estado atrás ON	CUMPLE
	4	Orden atrás (cuando tiene la orden dada)	Ninguna acción	CUMPLE
	5	Detección posición inicial	Estado PINI ON	CUMPLE
	6	Detección posición media	Estado MIDDLE ON	CUMPLE

	7	Detección posición final	Estado PEND ON	CUMPLE
	8	Estado adelante 5s y no se detecta PMIDDLE	Alarma MIDDLE ON	CUMPLE
	9	Estado atrás 5s y no se detecta PEND	Alarma PEND ON	CUMPLE
	10	Alarmas elemento ON	Elementos bloqueados	CUMPLE
	11	Orden <i>reset</i> alarma del componente	Alarma se resetea siempre y cuando el estado que las acciona ya no este activo	CUMPLE
F203_CY01	1	Orden de abrir (cuando no esta abierto)	Estado de apertura ON	CUMPLE
	2	Orden de abrir (cuando esta abierto)	Estado de apertura OFF	CUMPLE
	3	Detector final de carrera abierto	Estado FCO ON	CUMPLE
	4	Detector final de carrera cerrado	Estado FCT ON	CUMPLE
	5	Estado de apertura activo durante 5s sin detección de FCO	Alarma FCO ON	CUMPLE
	6	Estado de apertura OFF durante más de 5s segundo y FCT inactivo	Alarma FCT ON	CUMPLE
	7	Alarma Elemento ON	Elementos bloqueados	CUMPLE
	8	Orden <i>reset</i> alarma del componente	Alarma se resetea siempre y cuando el estado que las acciona ya no este activo	CUMPLE
F203_LS01	1	Detección pieza metálica	Estado detección ON	CUMPLE
F203_GP01	1	Orden apertura (cuando no esta abierto)	Estado de apertura ON	CUMPLE
	2	Orden de apertura (cuando ya esta abierto)	Ninguna acción	CUMPLE
	3	Orden de cierre (cuando no esta cerrado)	Estado de cierre ON	CUMPLE
	4	Orden de cierre (cuando está cerrado)	Ninguna acción	CUMPLE

Tabla 3.22. Pruebas realizadas a la FAS203.

A continuación, se muestran los resultados obtenidos de la FAS 204:

Elemento	Número	Prueba	Resultado esperado	Valoración
F204	1	Cuando la célula entra en remoto entra al estado por defecto	Modo manual	CUMPLE
	2	Intento de cambiar de modo mientras el sistema esta en alarma	Se puede cambiar de modo en circunstancias de alarma.	CUMPLE
	3	Modo en el que el sistema deja de estar en remoto	No hay ningún modo en el que deje de estar en remoto.	CUMPLE
	4	Activación alarma de un elemento	Activación alarma del sistema	CUMPLE
	5	Activación <i>reset</i> del sistema	Se activa el reseteo de la alarma del sistema siempre y cuando no haya ninguna alarma de componente activa	CUMPLE
	6	Sistema en alarma	No funciona la activación de ningún elemento.	CUMPLE
	7	Sistema en alarma y modo de mantenimiento activado	Es posible poder iniciar condiciones iniciales y activar actuadores	CUMPLE
F204_RA01	1	Orden adelante (mientras no esta activado)	Estado adelante ON	CUMPLE
	2	Orden adelante (mientras esta activado)	Estado adelante OFF	CUMPLE
	3	Detección posición final	Estado PEND ON	CUMPLE
	4	Detección posición inicial	Estado PINI ON	CUMPLE
	5	Estado adelante 5s y no se detecta PEND	Alarma ON	CUMPLE
	6	Alarma elemento ON	Elementos bloqueados	CUMPLE
	7	Orden <i>reset</i> alarma del componente	Alarma se resetea siempre y cuando el estado que las acciona ya no este activo	CUMPLE
F204_CY01	1	Orden de abrir (cuando no esta abierto)	Estado de abrir ON	CUMPLE

	2	Orden de abrir (cuando esta abierto)	Ninguna acción	CUMPLE
	3	Orden de cerrar (cuando no esta cerrado)	Estado de cerrar ON	CUMPLE
	4	Orden de cerrar (cuando esta cerrado)	Ninguna acción	CUMPLE
	5	Detector final de carrera abierto	Estado FCO ON	CUMPLE
	6	Detector final de carrera cerrado	Estado FCT ON	CUMPLE
	7	Estado de apertura activo durante 5s sin detección de FCO	Alarma FCO ON	CUMPLE
	8	Estado de apertura OFF durante más de 5s segundo y FCT inactivo	Alarma FCT ON	CUMPLE
	9	Alarma Elemento ON	Elementos bloqueados	CUMPLE
	10	Orden <i>reset</i> alarma del componente	Alarma se resetea siempre y cuando el estado que las acciona ya no este activo	CUMPLE
F204_CY02	1	Orden de abrir (cuando no esta abierto)	Estado de apertura ON	CUMPLE
	2	Orden de abrir (cuando esta abierto)	Estado de apertura OFF	CUMPLE
F204_CY03	1	Orden de abrir (cuando no esta abierto)	Estado de apertura ON	CUMPLE
	2	Orden de abrir (cuando esta abierto)	Estado de apertura OFF	CUMPLE
F204_CY04	1	Orden de abrir (cuando no esta abierto)	Estado de apertura ON	CUMPLE
	2	Orden de abrir (cuando esta abierto)	Estado de apertura OFF	CUMPLE
	3	Detector final de carrera abierto	Estado FCO ON	CUMPLE
	4	Detector final de carrera cerrado	Estado FCT ON	CUMPLE
	5	Estado de apertura activo durante 5s sin detección de FCO	Alarma FCO ON	CUMPLE

	6	Estado de apertura OFF durante más de 5s segundo y FCT inactivo	Alarma FCT ON	CUMPLE
	7	Alarma Elemento ON	Elementos bloqueados	CUMPLE
	8	Orden <i>reset</i> alarma del componente	Alarma se resetea siempre y cuando el estado que las acciona ya no este activo	CUMPLE
F204_AS01	1	Detección de variación de la variable cuando el potenciómetro cambia de altura	Variable cambia de valores con una escala de unidades	CUMPLE
F204_GP01	1	Orden apertura (cuando no esta abierto)	Estado de apertura ON	CUMPLE
	2	Orden de apertura (cuando ya esta abierto)	Ninguna acción	CUMPLE
	3	Orden de cierre (cuando no esta cerrado)	Estado de cierre ON	CUMPLE
	4	Orden de cierre (cuando está cerrado)	Ninguna acción	CUMPLE
FBELT_LS34	1	Detección pieza metálica	Estado detección ON	CUMPLE
FBELT_CY34	1	Orden de abrir (cuando no esta abierto)	Estado de apertura ON	CUMPLE
	2	Orden de abrir (cuando esta abierto)	Estado de apertura OFF	CUMPLE
FBELT_CY34UP	1	Orden de abrir (cuando no esta abierto)	Estado de apertura ON	CUMPLE
	2	Orden de abrir (cuando esta abierto)	Estado de apertura OFF	CUMPLE

Tabla 3.23. Pruebas realizadas a la FAS204.

A continuación, se muestran los resultados obtenidos de la FAS207:

Elemento	Número	Prueba	Resultado esperado	Valoración
F207	1	Cuando la célula entra en remoto entra al estado por defecto	Modo manual	CUMPLE

	2	Intento de cambiar de modo mientras el sistema esta en alarma	Se puede cambiar de modo en circunstancias de alarma.	CUMPLE
	3	Modo en el que el sistema deja de estar en remoto	No hay ningún modo en el que deje de estar en remoto.	CUMPLE
	4	Activación alarma de un elemento	Activación alarma del sistema	CUMPLE
	5	Activación <i>reset</i> del sistema	Se activa el reseteo de la alarma del sistema siempre y cuando no haya ninguna alarma de componente activa	CUMPLE
	6	Sistema en alarma	No funciona la activación de ningún elemento.	CUMPLE
	7	Sistema en alarma y modo de mantenimiento activado	Es posible poder iniciar condiciones iniciales y activar actuadores	CUMPLE
F207_CY01	1	Orden de abrir (cuando no esta abierto)	Estado de apertura ON	CUMPLE
	2	Orden de abrir (cuando esta abierto)	Estado de apertura OFF	CUMPLE
F207_CY02	1	Orden de abrir (cuando no esta abierto)	Estado de apertura ON	CUMPLE
	2	Orden de abrir (cuando esta abierto)	Estado de apertura OFF	CUMPLE
	3	Detector final de carrera abierto	Estado FCO ON	CUMPLE
	4	Detector final de carrera cerrado	Estado FCT ON	CUMPLE
	5	Estado de apertura activo durante 5s sin detección de FCO	Alarma FCO ON	CUMPLE
	6	Estado de apertura OFF durante más de 5s segundo y FCT inactivo	Alarma FCT ON	CUMPLE
	7	Alarma Elemento ON	Elementos bloqueados	CUMPLE
	8	Orden reset alarma del componente	Alarma se resetea siempre y cuando el estado que las acciona ya no este activo	CUMPLE

F207_CY03	1	Orden de abrir (cuando no esta abierto)	Estado de apertura ON	CUMPLE
	2	Orden de abrir (cuando esta abierto)	Estado de apertura OFF	CUMPLE
	3	Detector final de carrera abierto	Estado FCO ON	CUMPLE
	4	Detector final de carrera cerrado	Estado FCT ON	CUMPLE
	5	Estado de apertura activo durante 5s sin detección de FCO	Alarma FCO ON	CUMPLE
	6	Estado de apertura OFF durante más de 5s segundo y FCT inactivo	Alarma FCT ON	CUMPLE
	7	Alarma Elemento ON	Elementos bloqueados	CUMPLE
	8	Orden <i>reset</i> alarma del componente	Alarma se resetea siempre y cuando el estado que las acciona ya no este activo	CUMPLE
F207_CY04	1	Orden de abrir (cuando no esta abierto)	Estado de apertura ON	CUMPLE
	2	Orden de abrir (cuando esta abierto)	Estado de apertura OFF	CUMPLE
F207_IS01	1	Detección pieza metálica	Estado detección ON	CUMPLE
F207_CS01	1	Detección pieza	Estado detección ON	CUMPLE

Tabla 3.24. Pruebas realizadas a la FAS207.

A continuación, se muestran los resultados obtenidos de la FAS208:

Elemento	Número	Prueba	Resultado esperado	Valoración
F208	1	Cuando la célula entra en remoto entra al estado por defecto	Modo manual	CUMPLE
	2	Intento de cambiar de modo mientras el sistema esta en alarma	Se puede cambiar de modo en circunstancias de alarma.	CUMPLE
	3	Modo en el que el sistema deja de estar en remoto	No hay ningún modo en el que deje de estar en remoto.	CUMPLE
	4	Activación alarma de un elemento	Activación alarma del sistema	CUMPLE

	5	Activación <i>reset</i> del sistema	Se activa el reseteo de la alarma del sistema siempre y cuando no haya ninguna alarma de componente activa	CUMPLE
	6	Sistema en alarma	No funciona la activación de ningún elemento.	CUMPLE
	7	Sistema en alarma y modo de mantenimiento activado	Es posible poder iniciar condiciones iniciales y activar actuadores	CUMPLE
F208_RA01	1	Orden adelante (mientras no esta activado)	Estado adelante ON	CUMPLE
	2	Orden adelante (mientras esta activado)	Estado adelante OFF	CUMPLE
	3	Detección posición final	Estado PEND ON	CUMPLE
	4	Detección posición inicial	Estado PINI ON	NO CUMPLE
	5	Estado adelante 5s y no se detecta PEND	Alarma ON	CUMPLE
	6	Alarma elemento ON	Elementos bloqueados	CUMPLE
	7	Orden <i>reset</i> alarma del componente	Alarma se resetea siempre y cuando el estado que las acciona ya no este activo	CUMPLE
F208_CY01	1	Orden de abrir (cuando no esta abierto)	Estado de apertura ON	CUMPLE
	2	Orden de abrir (cuando esta abierto)	Estado de apertura OFF	CUMPLE
	3	Detector final de carrera abierto	Estado FCO ON	CUMPLE
	4	Detector final de carrera cerrado	Estado FCT ON	CUMPLE
	5	Estado de apertura activo durante 5s sin detección de FCO	Alarma FCO ON	CUMPLE
	6	Estado de apertura OFF durante más de 5s segundo y FCT inactivo	Alarma FCT ON	CUMPLE
	7	Alarma Elemento ON	Elementos bloqueados	CUMPLE

	8	Orden <i>reset</i> alarma del componente	Alarma se resetea siempre y cuando el estado que las acciona ya no este activo	CUMPLE
F208_CY02	1	Orden de abrir (cuando no esta abierto)	Estado de abrir ON	CUMPLE
	2	Orden de abrir (cuando esta abierto)	Ninguna acción	CUMPLE
	3	Orden de cerrar (cuando no esta cerrado)	Estado de cerrar ON	CUMPLE
	4	Orden de cerrar (cuando esta cerrado)	Ninguna acción	CUMPLE
	5	Detector final de carrera abierto	Estado FCO ON	CUMPLE
	6	Detector final de carrera cerrado	Estado FCT ON	CUMPLE
	7	Estado de apertura activo durante 5s sin detección de FCO	Alarma FCO ON	CUMPLE
	8	Estado de apertura OFF durante más de 5s segundo y FCT inactivo	Alarma FCT ON	CUMPLE
	9	Alarma Elemento ON	Elementos bloqueados	CUMPLE
	10	Orden <i>reset</i> alarma del componente	Alarma se resetea siempre y cuando el estado que las acciona ya no este activo	CUMPLE
F208_CY03	1	Orden de abrir (cuando no esta abierto)	Estado de apertura ON	CUMPLE
	2	Orden de abrir (cuando esta abierto)	Estado de apertura OFF	CUMPLE
	3	Detector final de carrera abierto	Estado FCO ON	CUMPLE
	4	Detector final de carrera cerrado	Estado FCT ON	CUMPLE
	5	Estado de apertura activo durante 5s sin detección de FCO	Alarma FCO ON	CUMPLE
	6	Estado de apertura OFF durante más de 5s segundo y FCT inactivo	Alarma FCT ON	CUMPLE

	7	Alarma Elemento ON	Elementos bloqueados	CUMPLE
	8	Orden <i>reset</i> alarma del componente	Alarma se resetea siempre y cuando el estado que las acciona ya no este activo	CUMPLE
F208_VS01	1	Orden de coger (cuando no esta cogiendo)	Estado de coger ON	CUMPLE
	2	Orden de coger (cuando esta cogiendo)	Ninguna acción	CUMPLE
	3	Orden de dejar (cuando no lo esta haciendo)	Estado de dejar ON	CUMPLE
	4	Orden de dejar (cuando no lo esta haciendo)	Ninguna acción	CUMPLE
	5	Detección de vacío	Estado OK ON	CUMPLE
	6	Estado de coger ON durante 5s y no se recibe señal del presostato	Alarma de componente ON	CUMPLE
	7	Alarma elemento ON	Elementos bloqueados	CUMPLE
	8	Orden <i>reset</i> Alarma elemento	Alarma elemento OFF	CUMPLE
F208_VS02	1	Orden de coger (cuando no esta cogiendo)	Estado de coger ON	CUMPLE
	2	Orden de coger (cuando esta cogiendo)	Ninguna acción	CUMPLE
	3	Orden de dejar (cuando no lo esta haciendo)	Estado de dejar ON	CUMPLE
	4	Orden de dejar (cuando no lo esta haciendo)	Ninguna acción	CUMPLE
	5	Detección de vacío	Estado OK ON	CUMPLE
	6	Estado de coger ON durante 5s y no se recibe señal del presostato	Alarma de componente ON	CUMPLE
	7	Alarma elemento ON	Elementos bloqueados	CUMPLE
	8	Orden <i>reset</i> Alarma elemento	Alarma elemento OFF	CUMPLE
FBELT_LS78	1	Detección pieza metálica	Estado detección ON	CUMPLE
FBELT_CY78	1	Orden de abrir (cuando no esta abierto)	Estado de apertura ON	CUMPLE
	2	Orden de abrir (cuando esta abierto)	Estado de apertura OFF	CUMPLE

Tabla 3.25. Pruebas realizadas en la FAS208.

5. Normativa

La normativa es un aspecto muy importante a tener en cuenta en el momento de la ejecuci3n del proyecto. Esto es de obligado cumplimiento si nuestro proyecto debe pasar alg3n tipo de auditorio mediante la cual poder adquirir una certificaci3n o certificado por el estilo. Por otro lado, tambi3n es interesante cumplir la normativa para que otros profesionales del sector sean capaces de poder entender nuestro proyecto y trabajar con 3l.

Se ha seguido un criterio y una normativa para cada parte del proyecto, a continuaci3n, se muestran y exponen dichas normativas.

5.1. Codificaci3n de los elementos

La normativa seguida en este caso es la ISA, concretamente la norma ISA-5.1-1984 (R1992). Esta norma normaliza la simbolog3a y la identificaci3n de la instrumentaci3n industrial.

Siguiendo este criterio, la nomenclatura del sistema se ha llevado a cabo con el siguiente criterio.

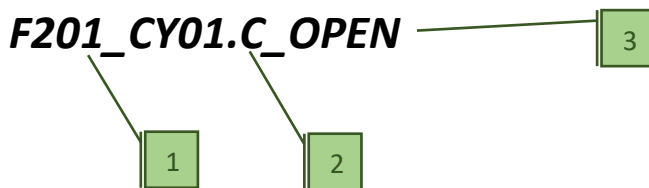


Fig.- 3.75. S3mbolo ISA.

N3mero	Descripci3n
1	Nombre principal del <i>tag</i> , la primera parte es el nombre de la FAS en la que se esta operando y la segunda parte es el componente, los nombres de los componentes como hemos visto en apartados anteriores han sido codificados, tambi3n siguiendo la norma que estamos nombrando.
2	Representa la funcionalidad del <i>tag</i> , la C representa una orden, la E representa un estado, la X representa una entrada al sistema y, por 3ltimo, la Y representa una entrada al sistema.
3	Descripci3n del bit que se esta manipulando.

Tabla 3.26. Definici3n de las partes del *tag*.

5.2. Programación PLC

En cuanto a la programación del PLC, hay una institución que posee un reglamento acerca de los autómatas programables, que es la IEC (*International Electrotechnical Commission*), más específicamente la norma IEC61131-3. La parte 3 de esta norma, establece los criterios de programación con los cuales son programados los autómatas, en nuestro caso nos interesa lo que hace referencia al lenguaje LD (*Ladder Diagram*) y SFC (*Secuencial Flow Chart*).

5.3. Programación aplicación SCADA

Como apoyo para normalizar el desarrollo de la aplicación SCADA se ha tomado como norma y ayuda la guía GUEDES, referente a la creación de interfaz de supervisión. Con ella nos hemos ayudado para establecer los colores de los componentes en sus diferentes estados, el color de las alarmas, la navegación por las pantallas y en la distribución de los botones y los componentes a los largos de las ventanas de la aplicación.

6. Conclusiones

Una vez acabado el proyecto, echando la vista atrás y viendo todo lo conseguido, se puede concluir que los objetivos ideados en un principio se han alcanzado con mérito. Estableciendo todos los objetivos que se han ido poniendo a lo largo de los meses de desarrollo del proyecto.

El hecho de que el proyecto se componga de pequeñas fases que se deben ir cumpliendo para poder avanzar a la siguiente, ha facilitado el poder ir avanzando paulatinamente y no tener muchos frentes abiertos a la vez, por el contrario, haber tenido las buenas prácticas de empezar un asunto y acabarlo para poder avanzar al siguiente.

Esto se puede ver reflejado en la codificación de los registros, en primer lugar, se ha tenido que asociar los registros reales a unos nuevos creados para que puedan ser manipulados por el controlador *front-end* esto ha ayudado a poder comprender mejor la asociación de los registros y la relación entre las entradas y salidas del sistema.

Se debe destacar que es importante el hecho de hacer un buen estudio previo del sistema, comprender sus partes, tener clara la función de cada componente, las entradas del sistema, sus salidas, etc. Un buen estudio inicial es la clave para desarrollar mejor fluidez a lo largo del desarrollo del proyecto.

En cuanto a la programación del controlador, ha sido muy satisfactoria debido a que los lenguajes utilizados son muy intuitivos. Inicialmente el SFC no parecía tan intuitivo como el *ladder* pero en cuanto a secuencias de automatización y procesos que deben seguir un orden de funcionamiento son muy útiles. El lenguaje SFC facilita el reconocimiento de errores o enclavamiento debido al no cumplimiento de condiciones, es un lenguaje muy fácil de rastrear.

La comunicación entre controladores aparentaba una dificultad mayor, pero, con el *software* adecuado se puede enlazar dos controladores mediante mensajería de manera rápida y sencilla.

Por último, la aplicación SCADA, representa una recreación de las tres parejas de células. Creando, en la medida de lo posible, un sinóptico fiel al entorno y fácil de comprender.

Para finalizar, se puede concluir de manera satisfactoria, habiendo cumplido todos los objetivos y logrando aspectos y características del sistema que se han ido implementando a medida que se ha desarrollado el proyecto. Durante el desarrollo del mismo se han ido adquiriendo unos conocimientos muy útiles para el futuro, como aprendizaje y como lecciones, en tareas de automatización industrial.

En definitiva, se ha cumplido el objetivo que se planteo en los inicios del proyecto.

7. Bibliografía

- [1] Allen-Bradley, «Rockwell Automation,» [En línia]. Available:
https://literature.rockwellautomation.com/idc/groups/literature/documents/td/1769-td005_-en-p.pdf.

- [2] Allen-Bradley, «Rockwell Automation,» [En línia]. Available:
https://literature.rockwellautomation.com/idc/groups/literature/documents/rm/1756-rm003_-es-p.pdf.

- [3] Allen-Bradley, «Rockwell Automation,» [En línia]. Available:
https://literature.rockwellautomation.com/idc/groups/literature/documents/pm/1756-pm006_-en-p.pdf.

- [4] Allen-Bradley, «Rockwell Automation,» [En línia]. Available:
https://literature.rockwellautomation.com/idc/groups/literature/documents/pm/1756-pm008_-en-p.pdf.

- [5] SMC_Manual_de_Usuario, «SMC International Training,» [En línia]. Available:
<https://www.smctraining.com/webpage/indexpage/235>.

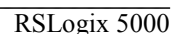
- [6] Guía_de_Usuario_In_Touch, «Wonderware,» [En línia]. Available:
<https://www.wonderware.es/HMI-SCADA/InTouch/>.

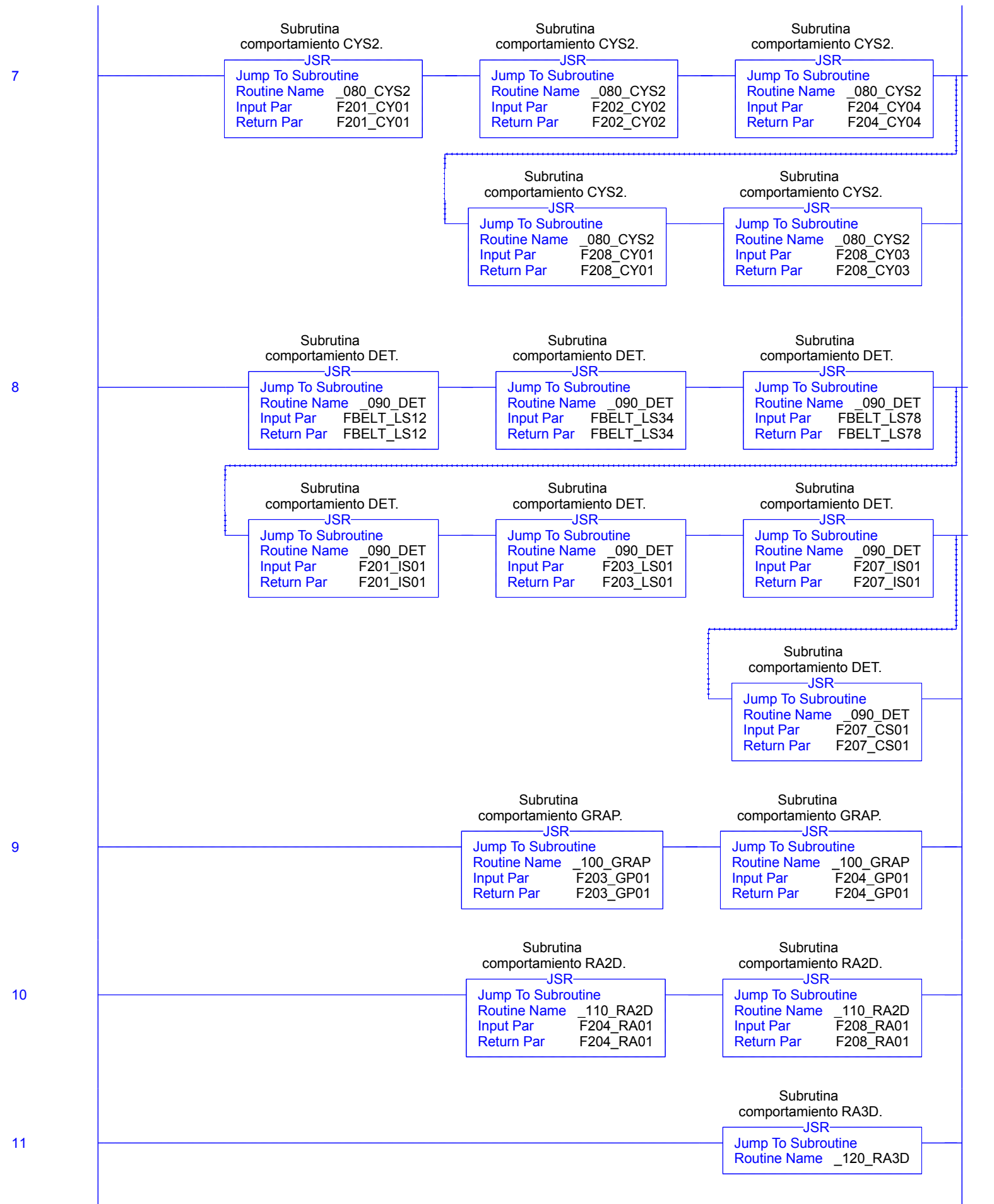
ANEXOS

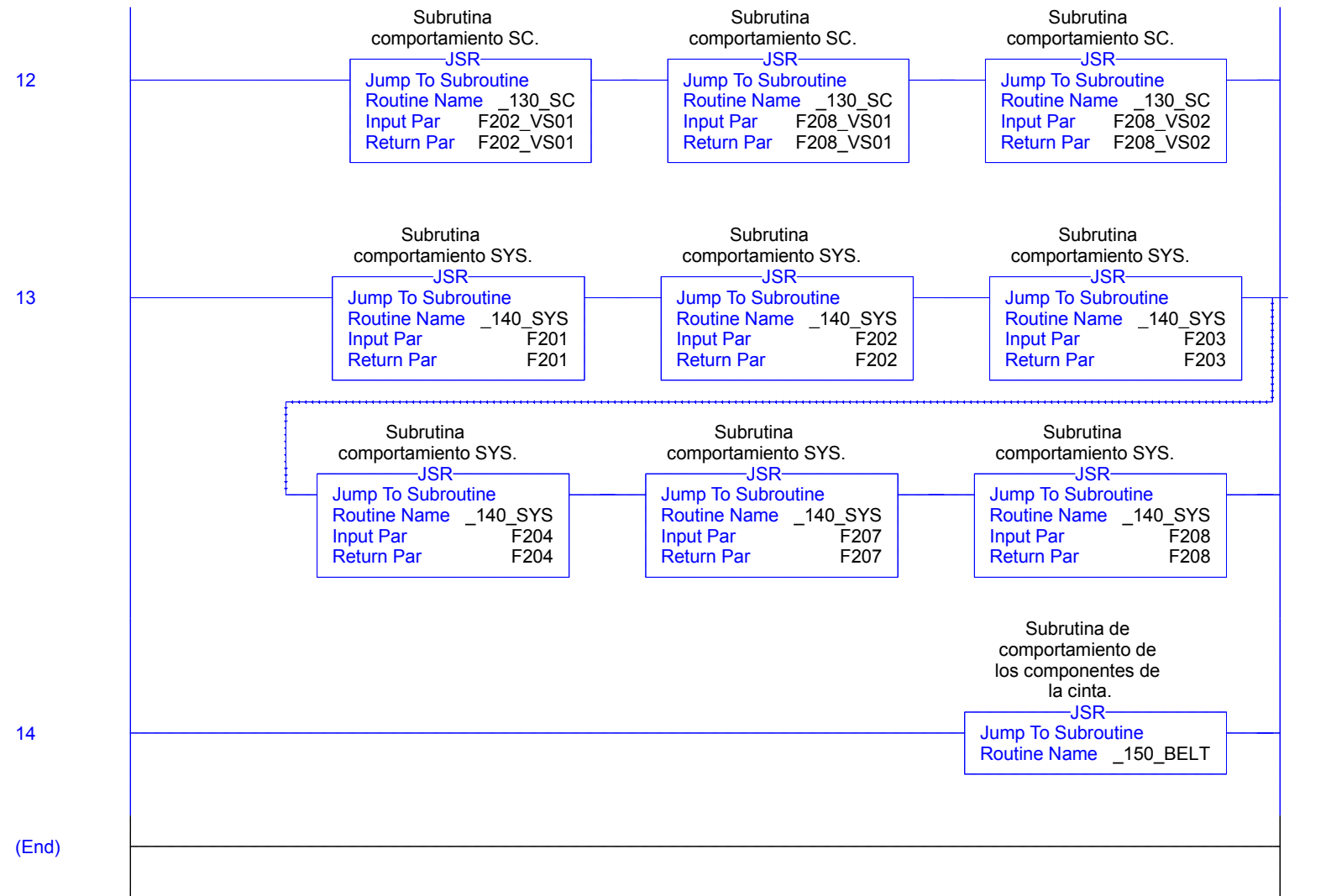
ANEXO 1:

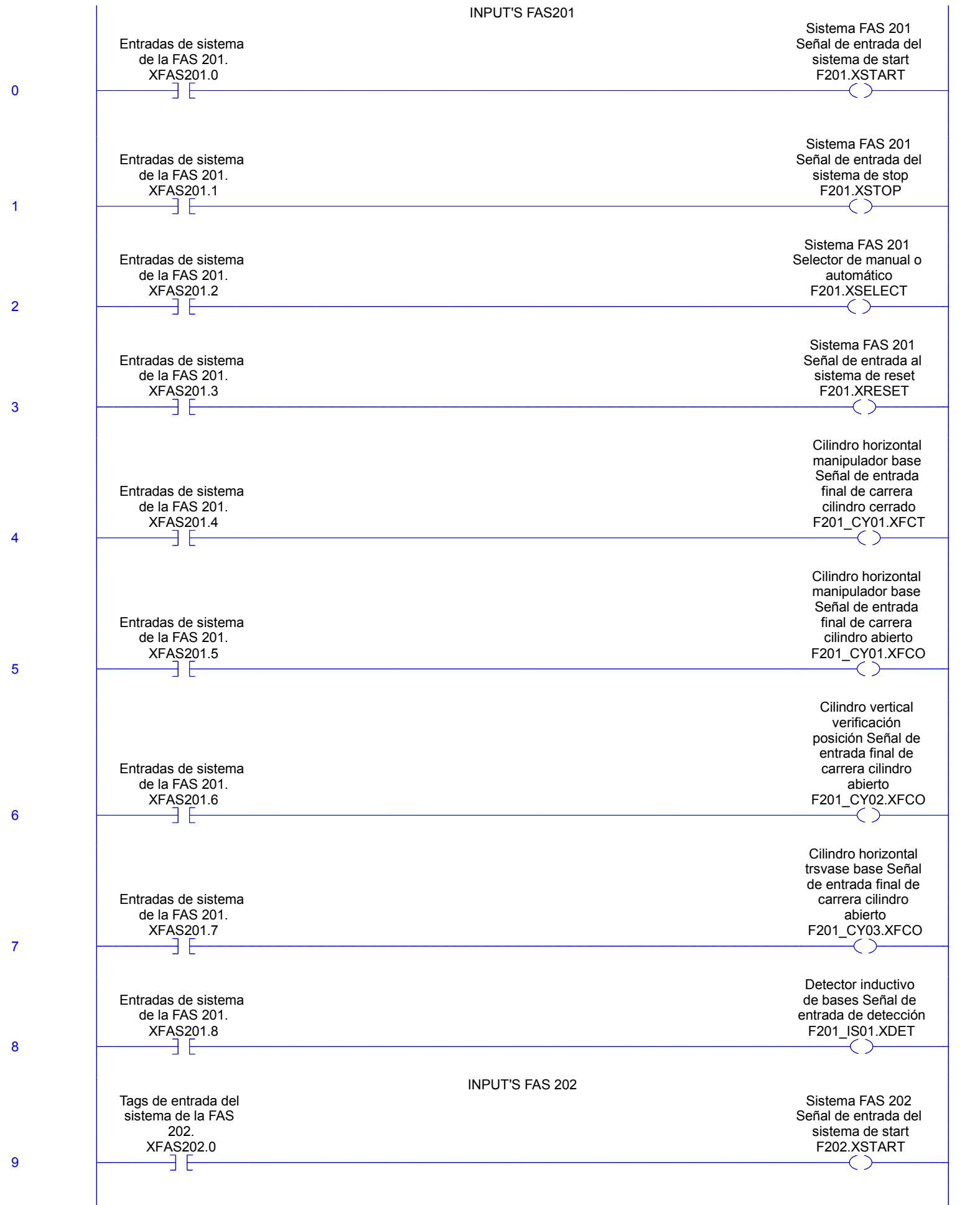
PROGRAMACI3N COMPONENTES

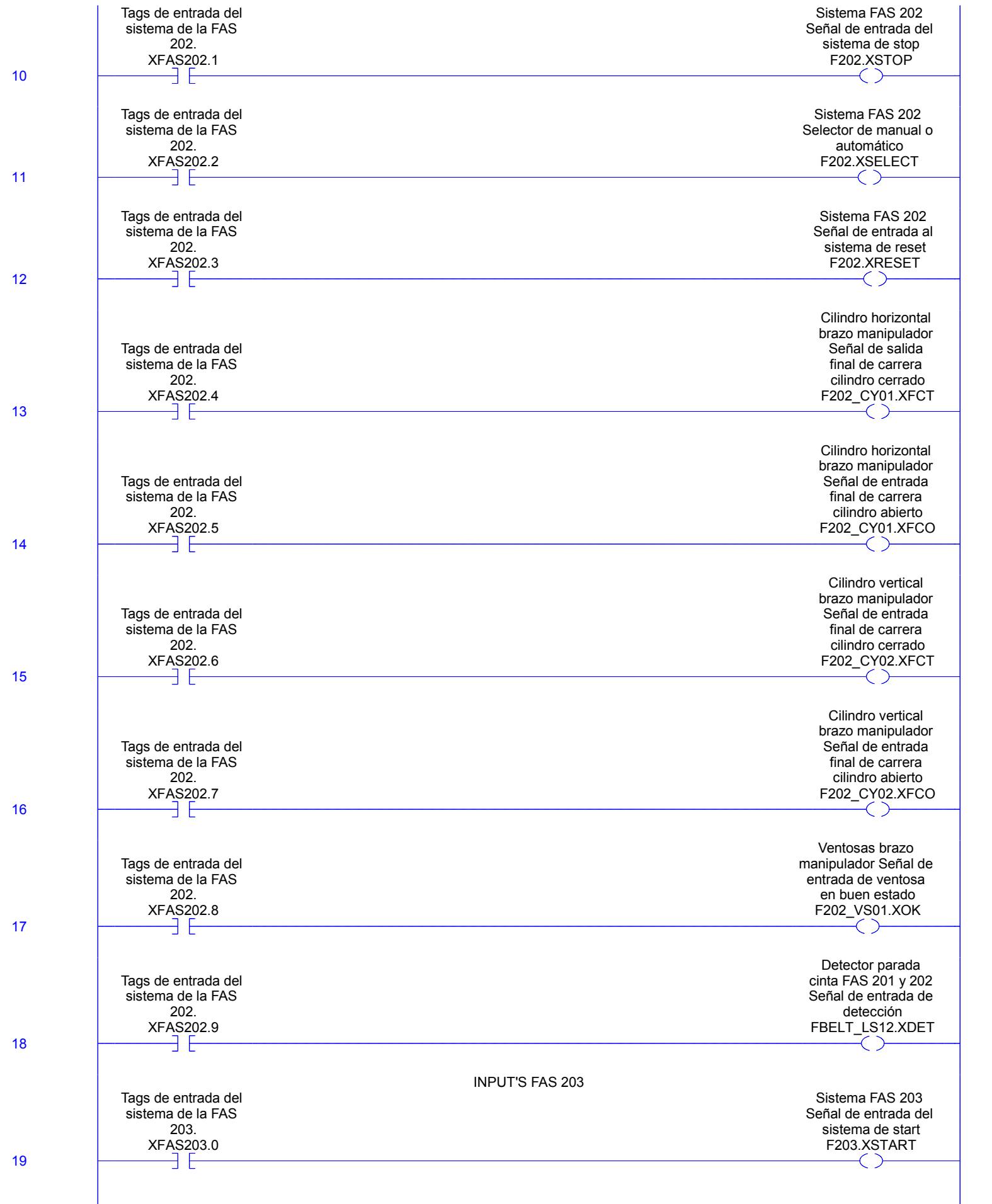
RSLogix 5000

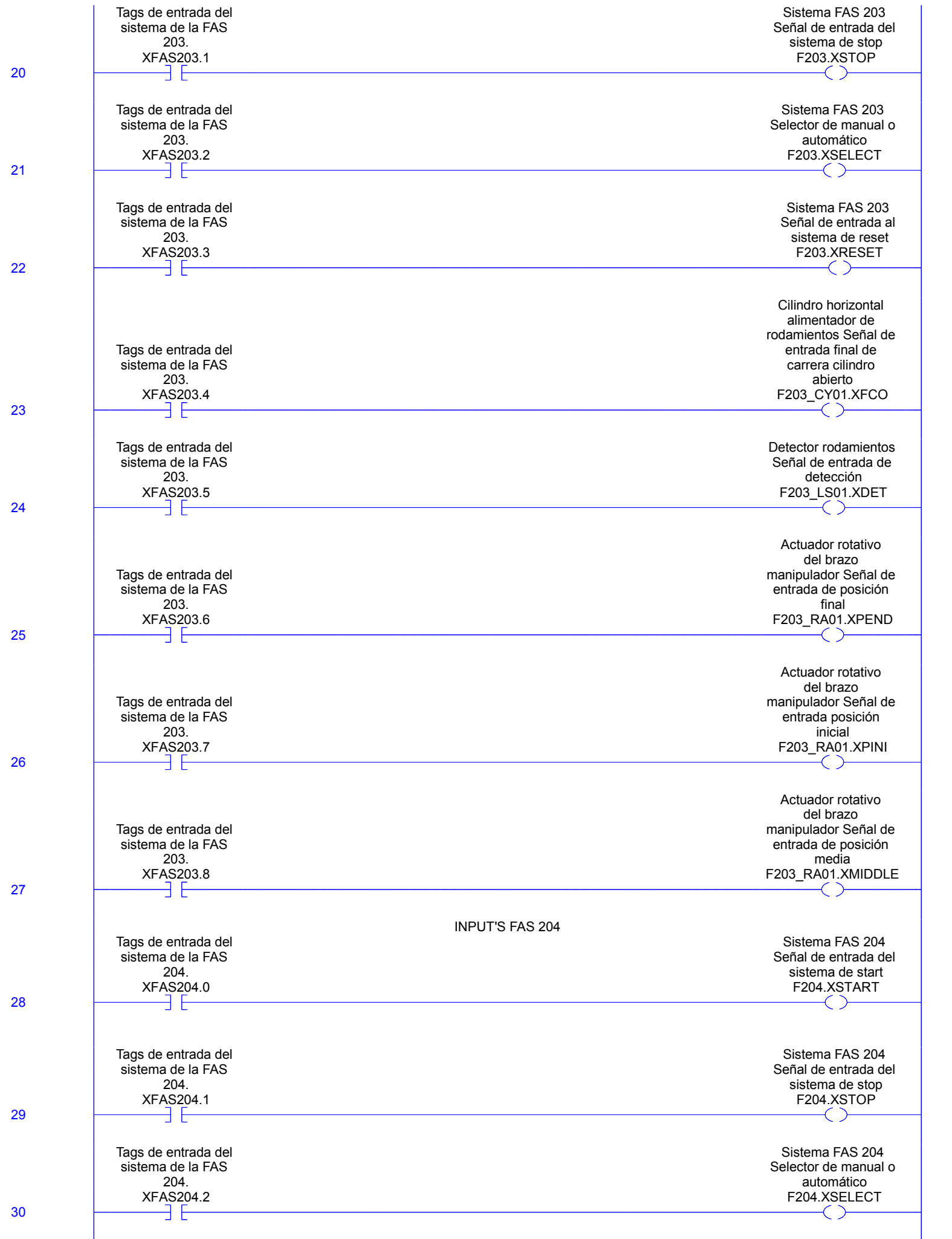


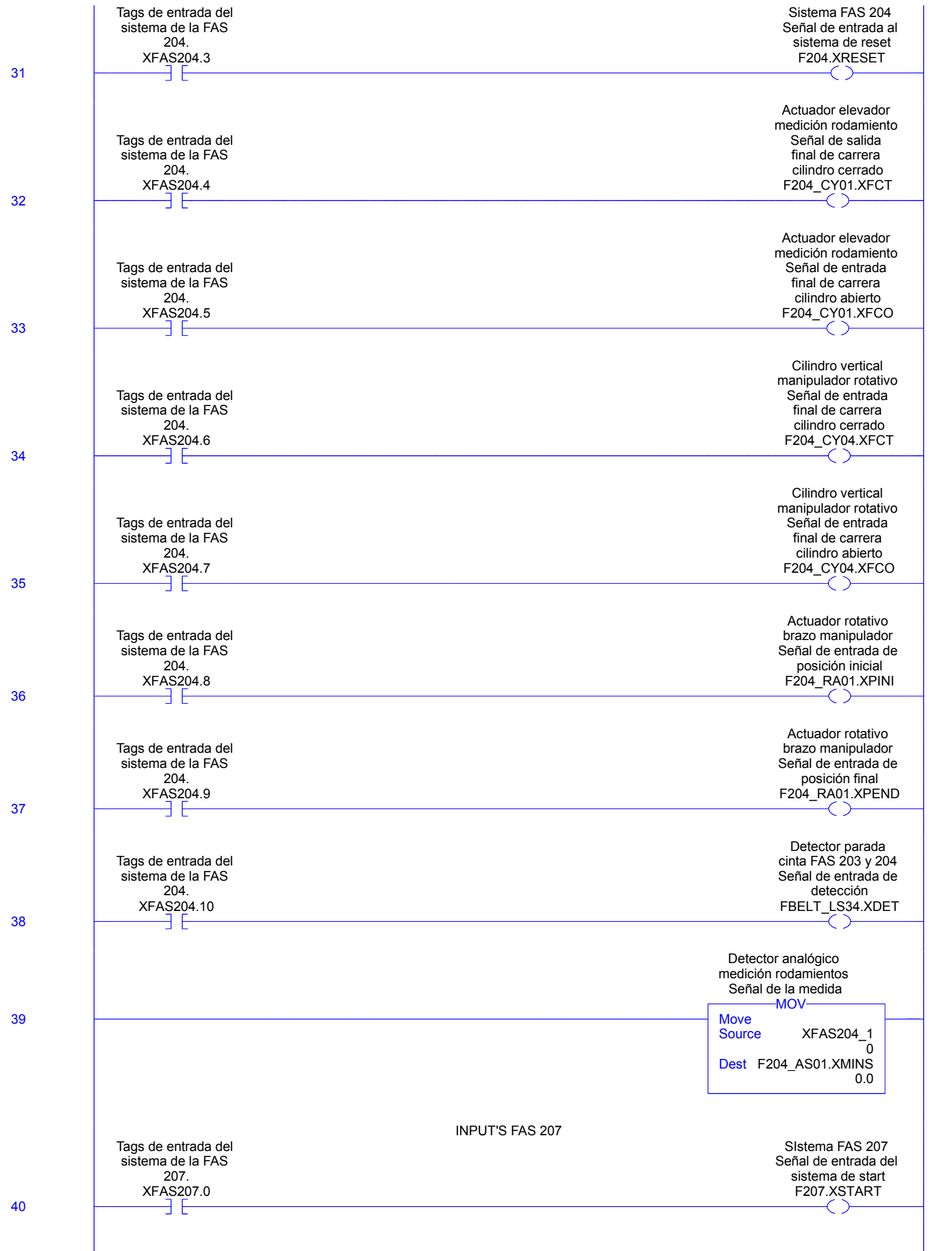




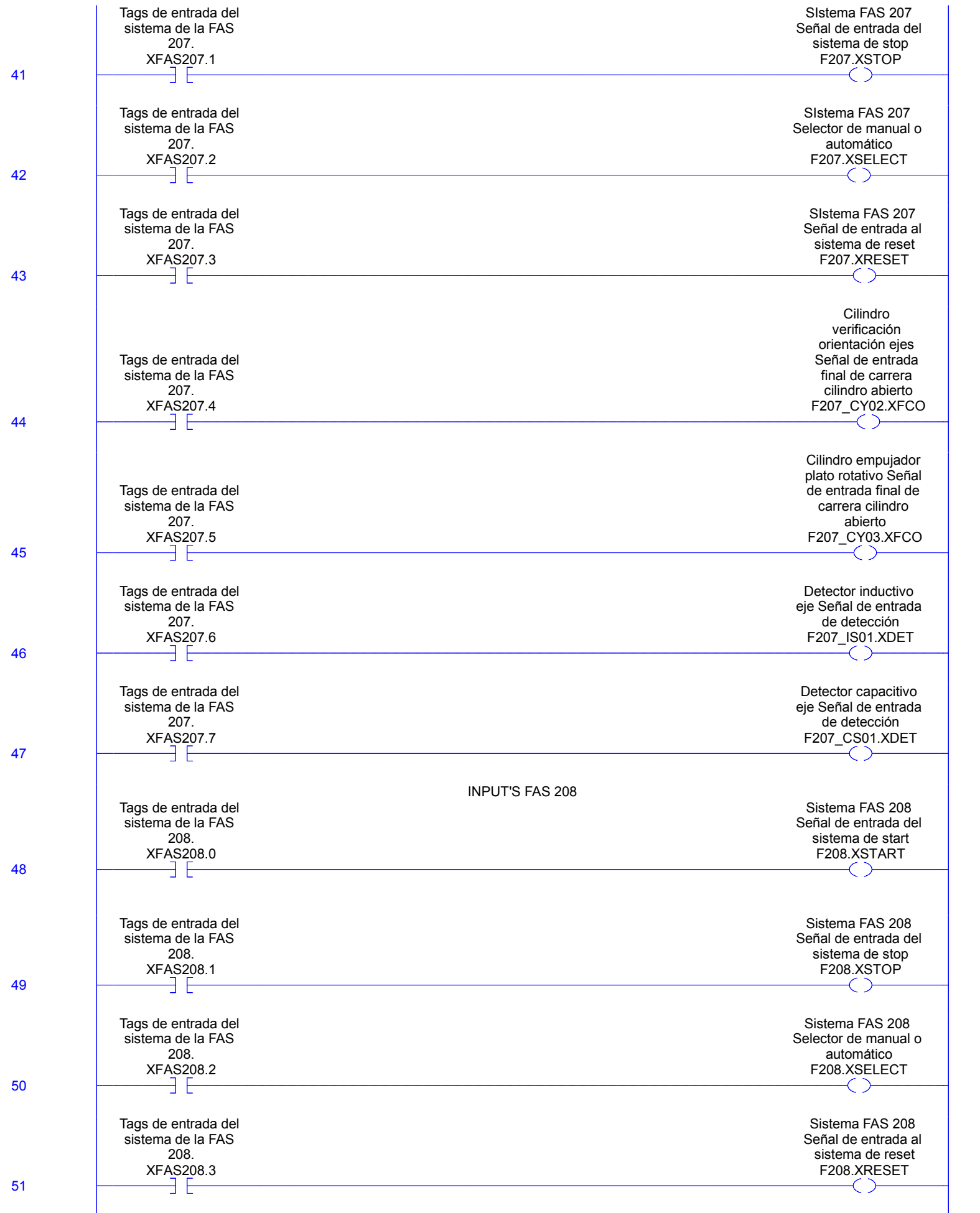


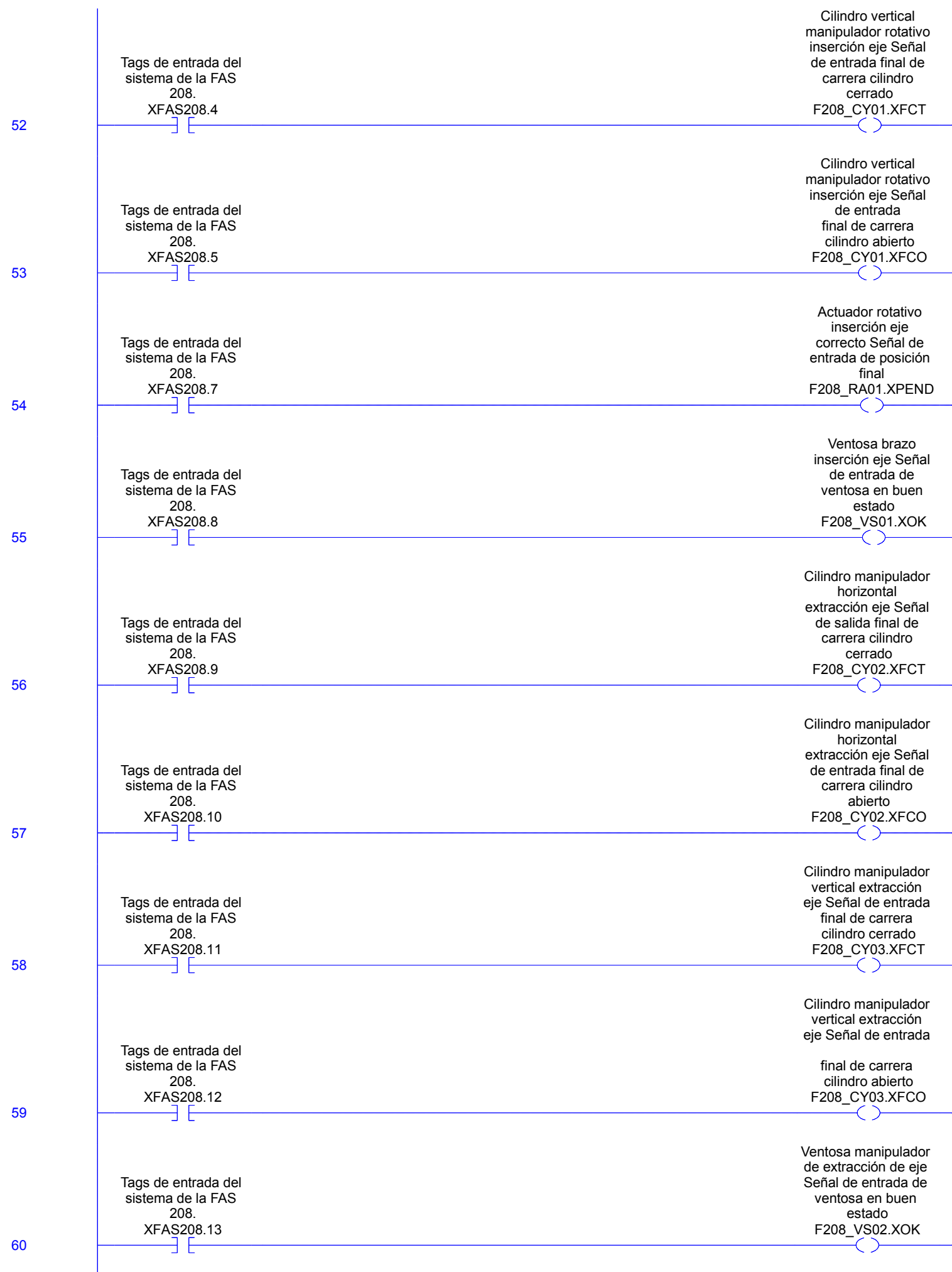


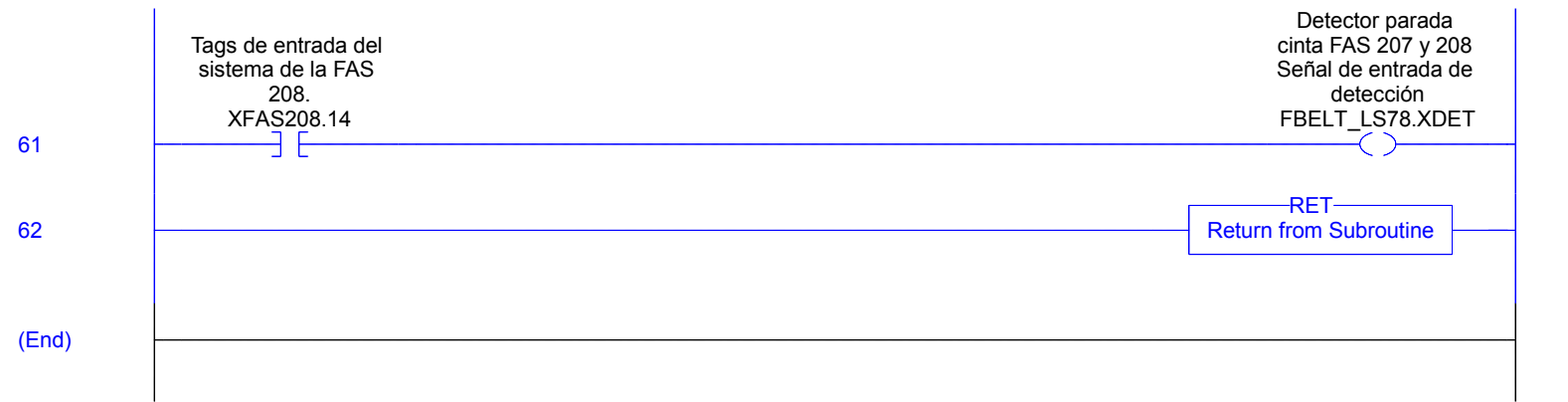


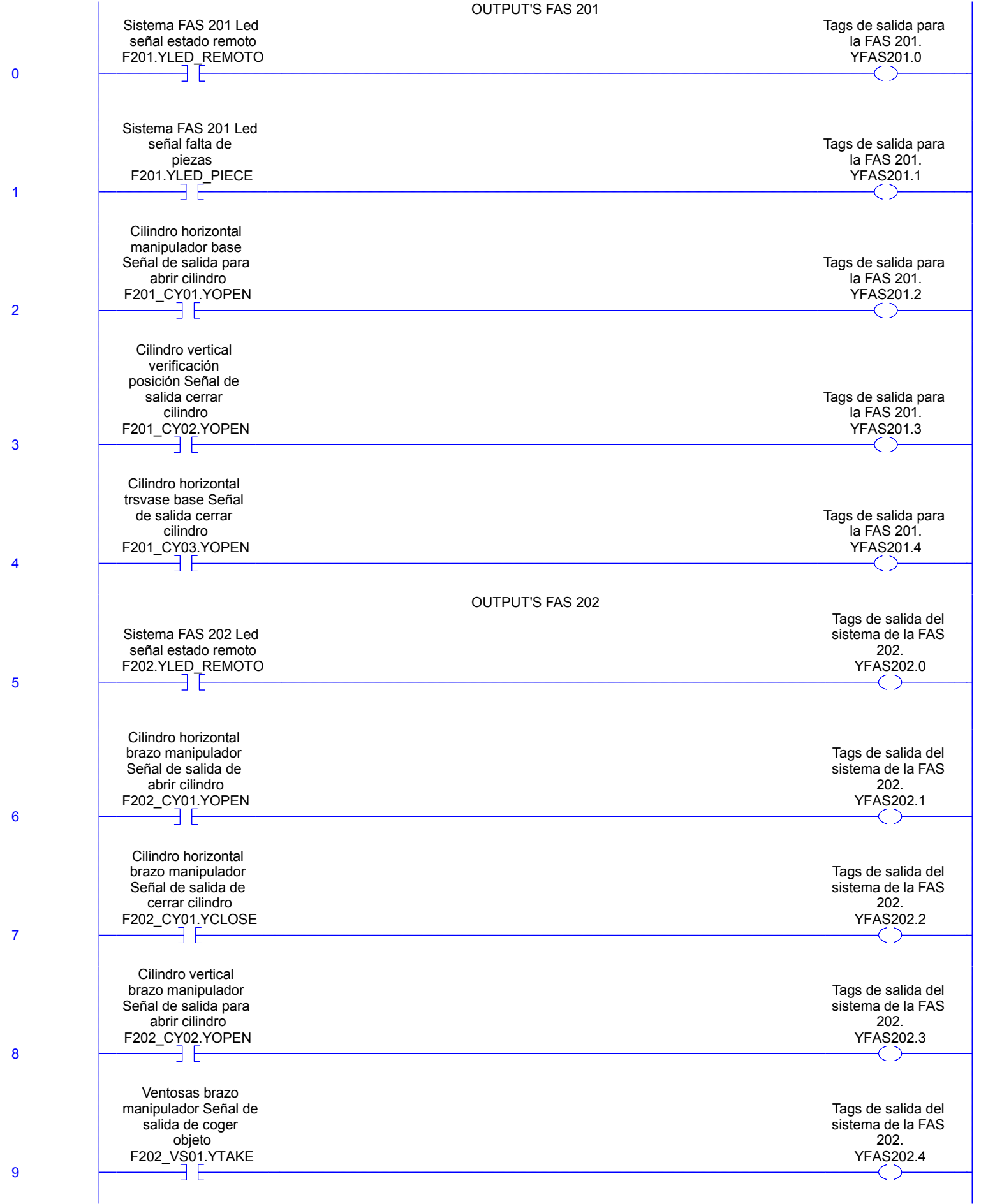


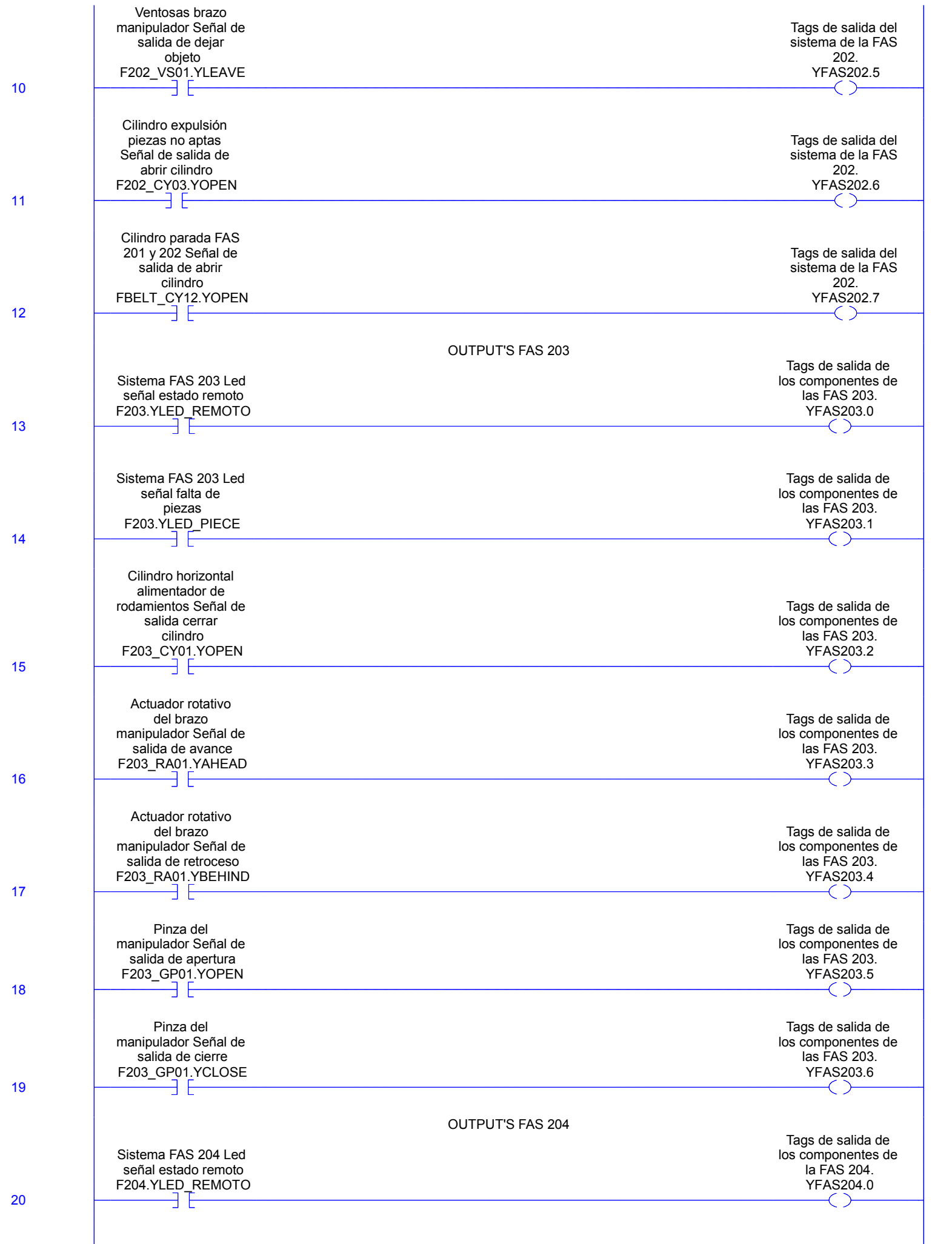
INPUT'S FAS 207

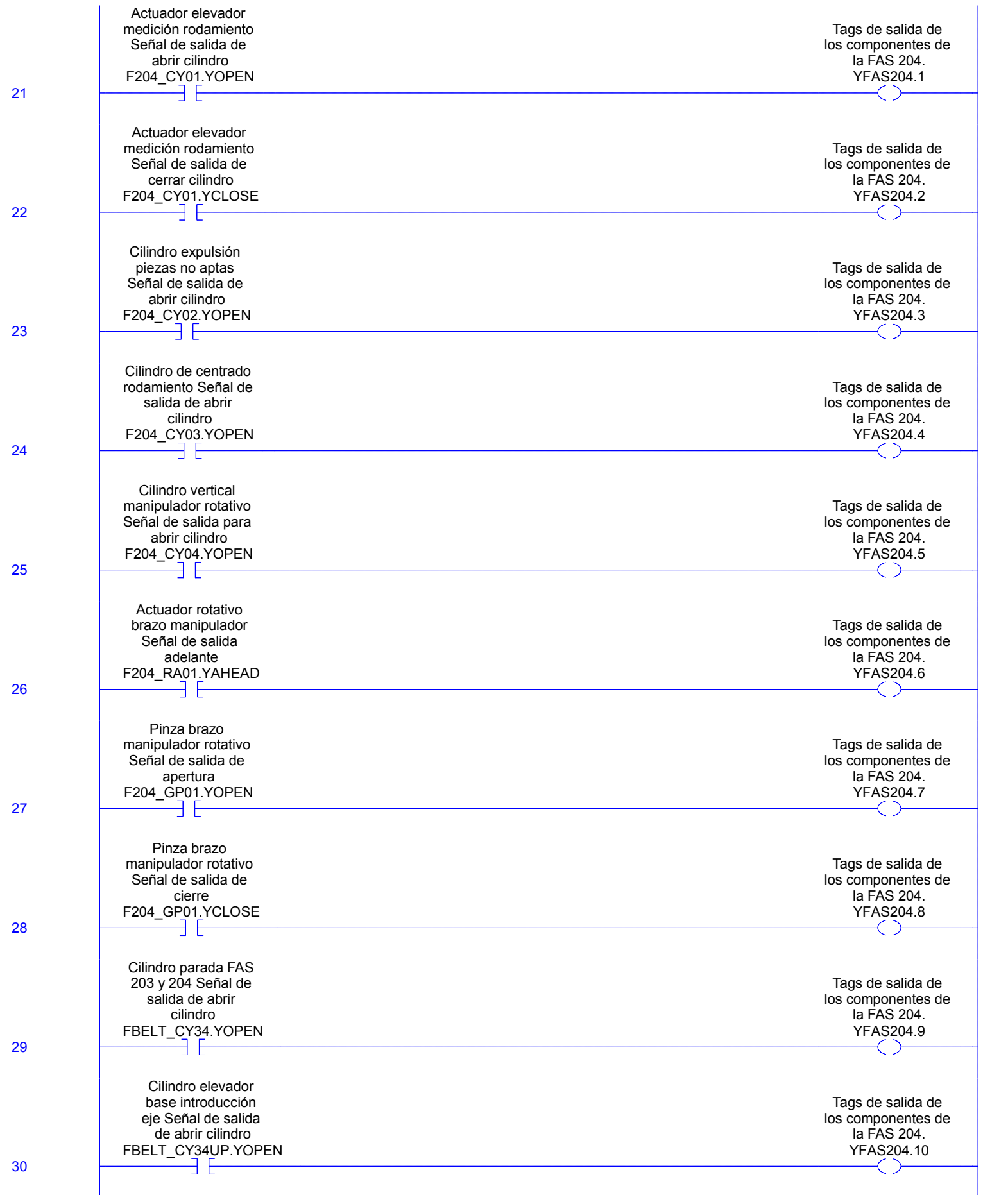


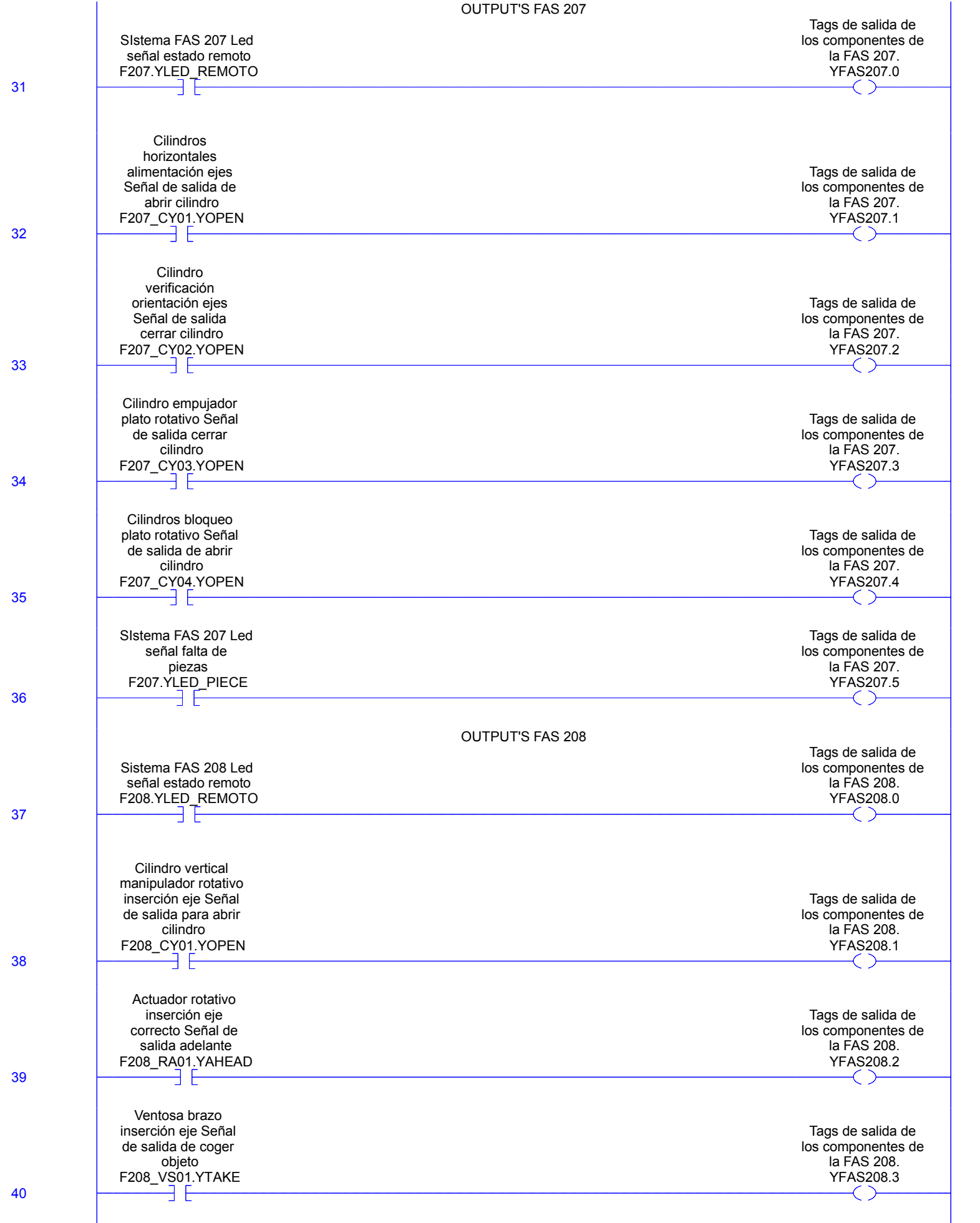


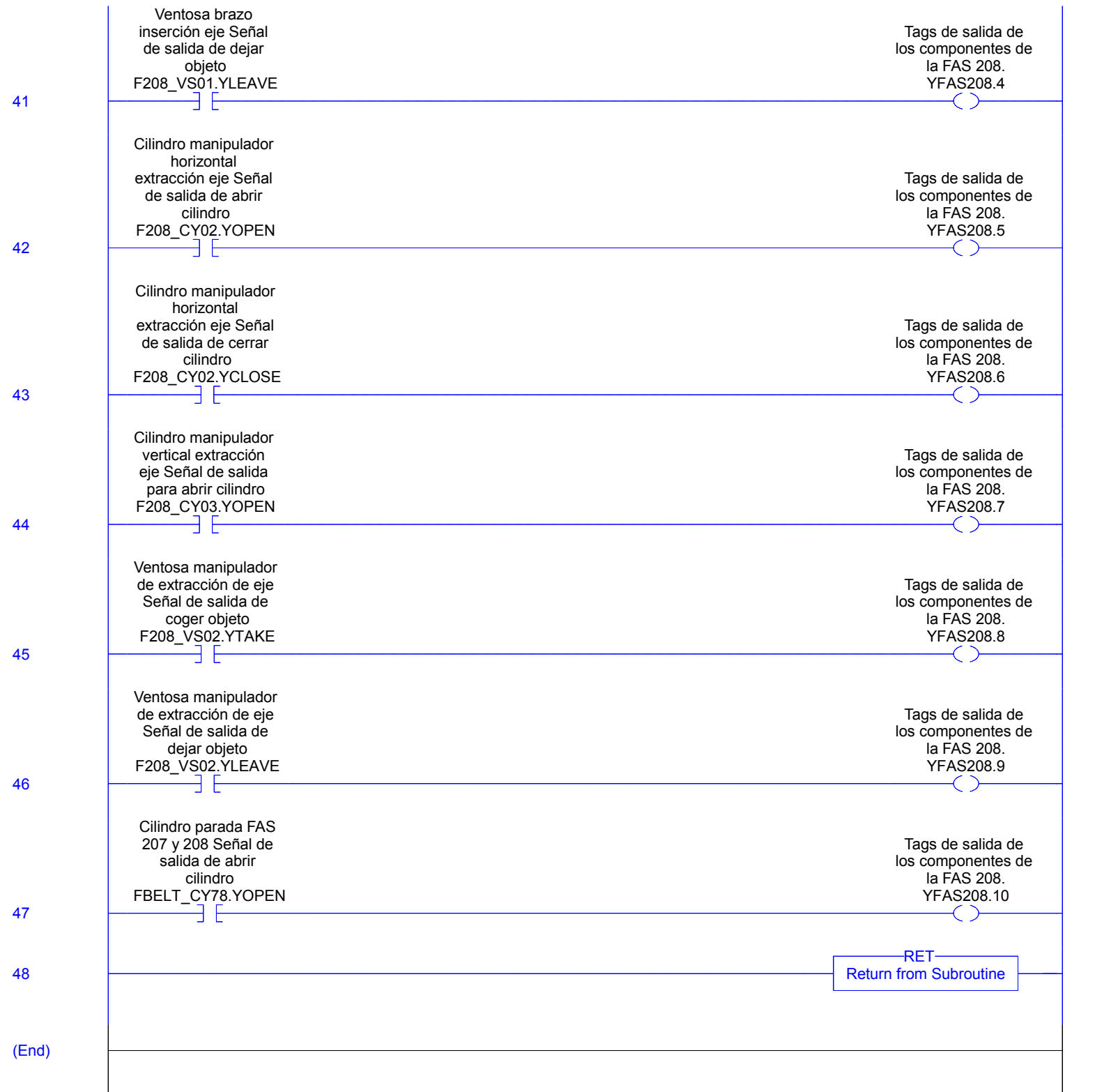


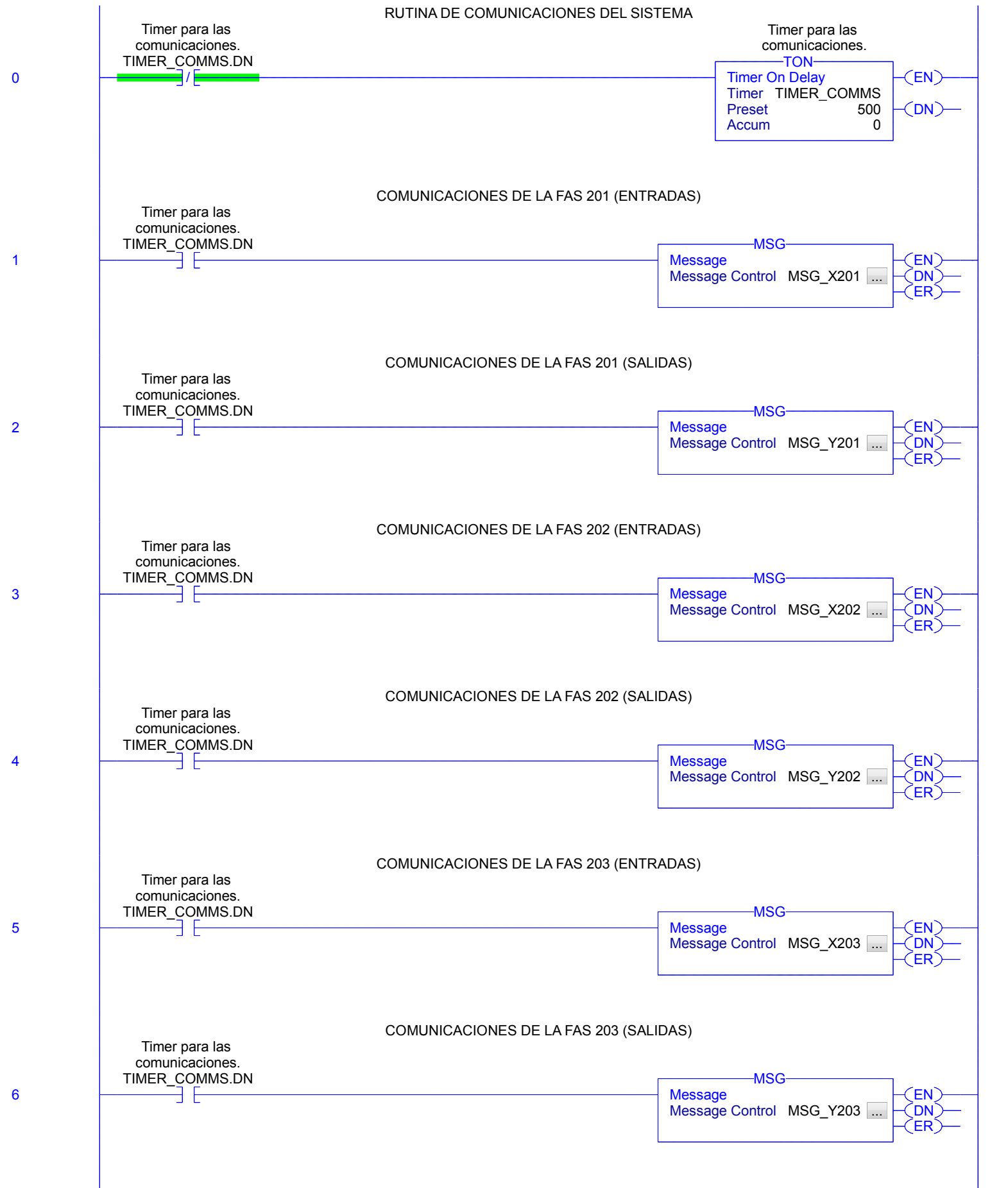


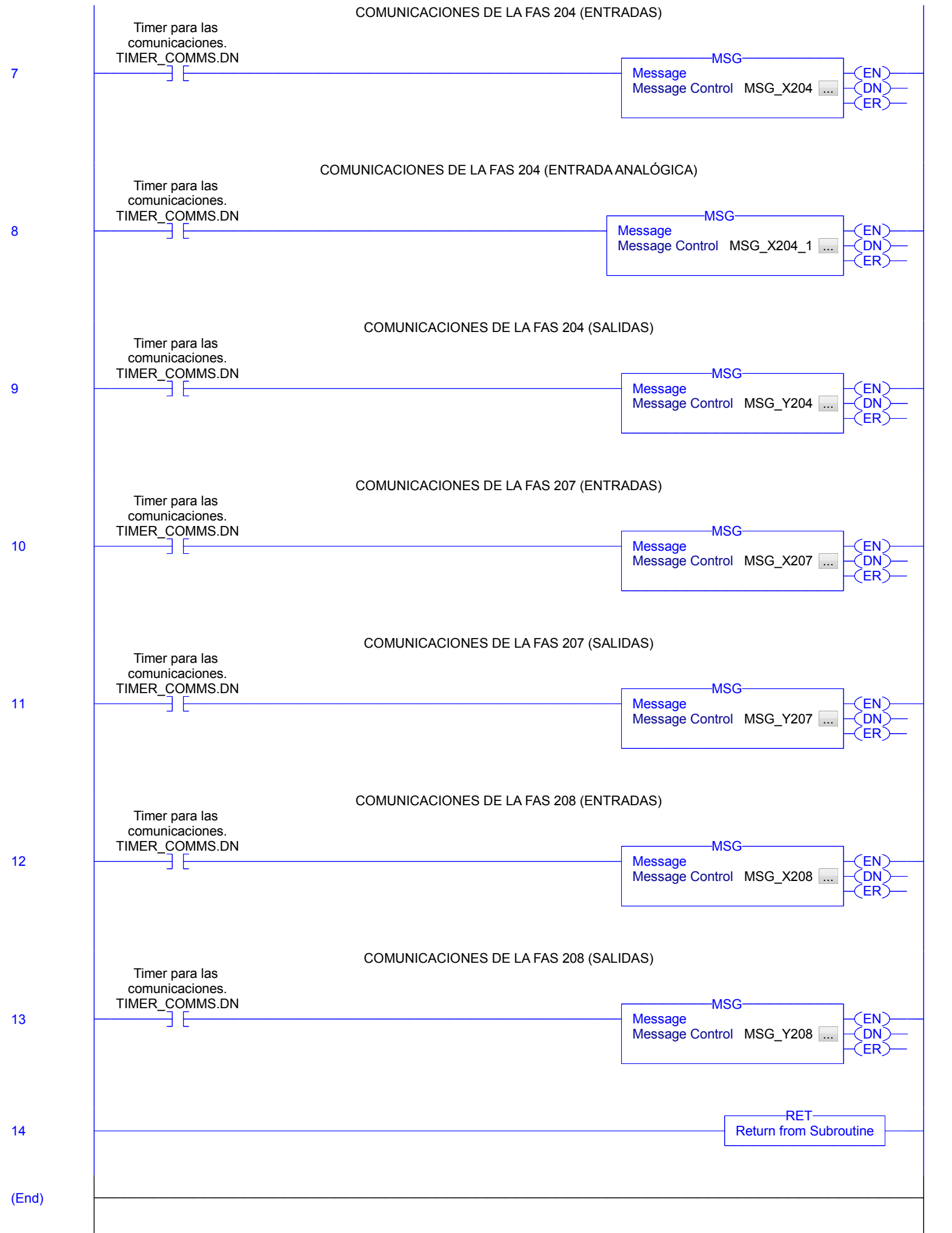


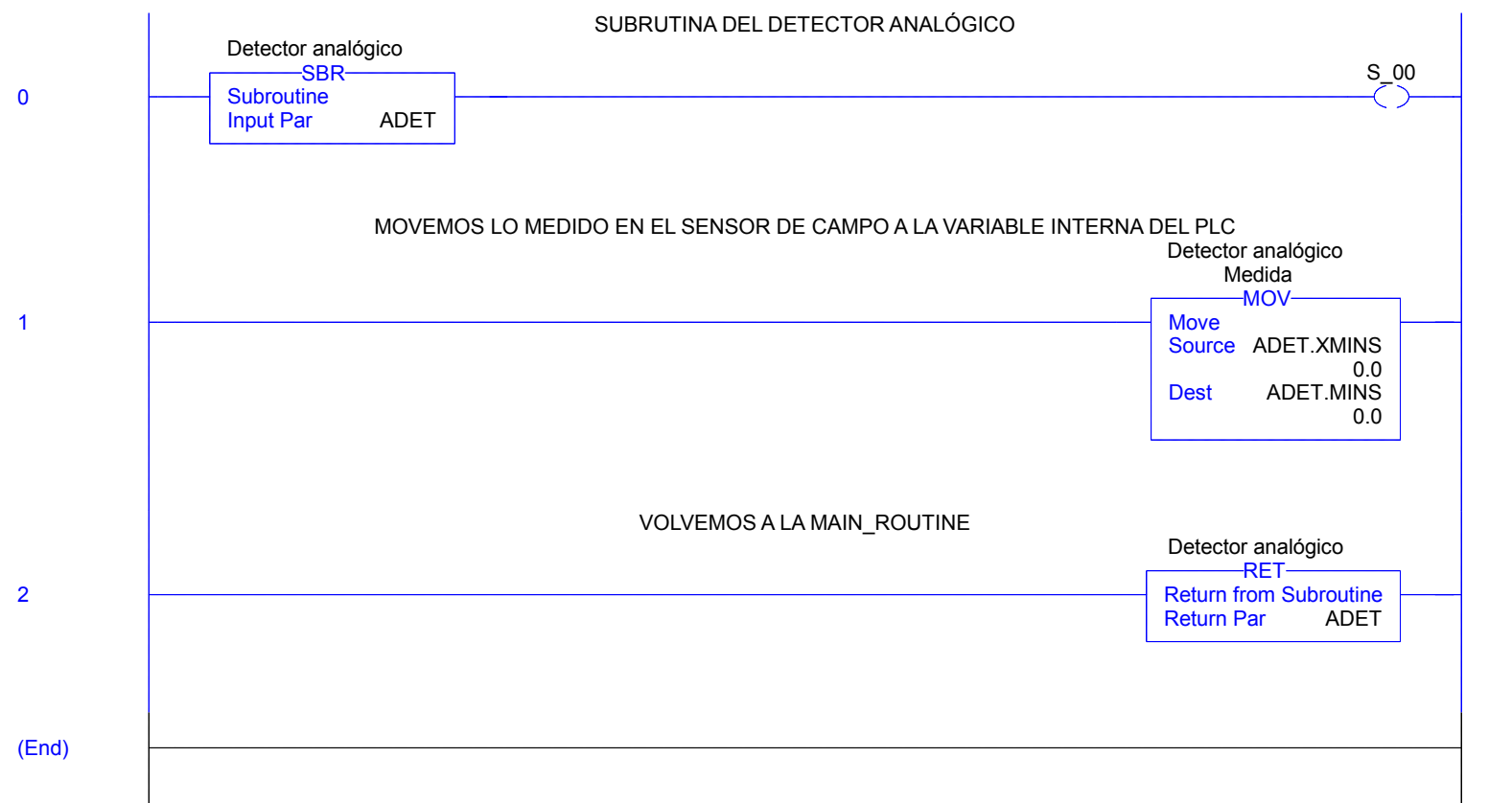


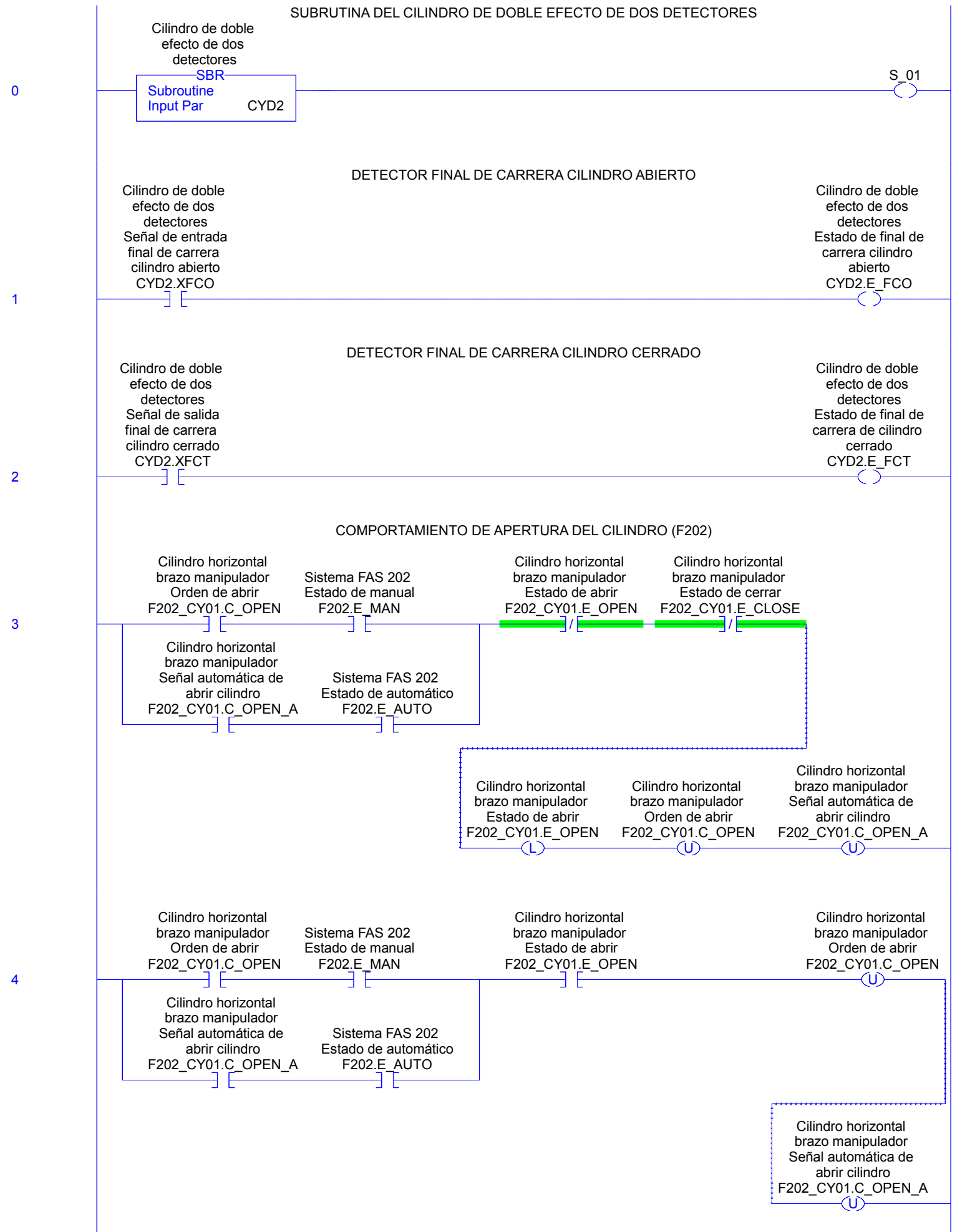


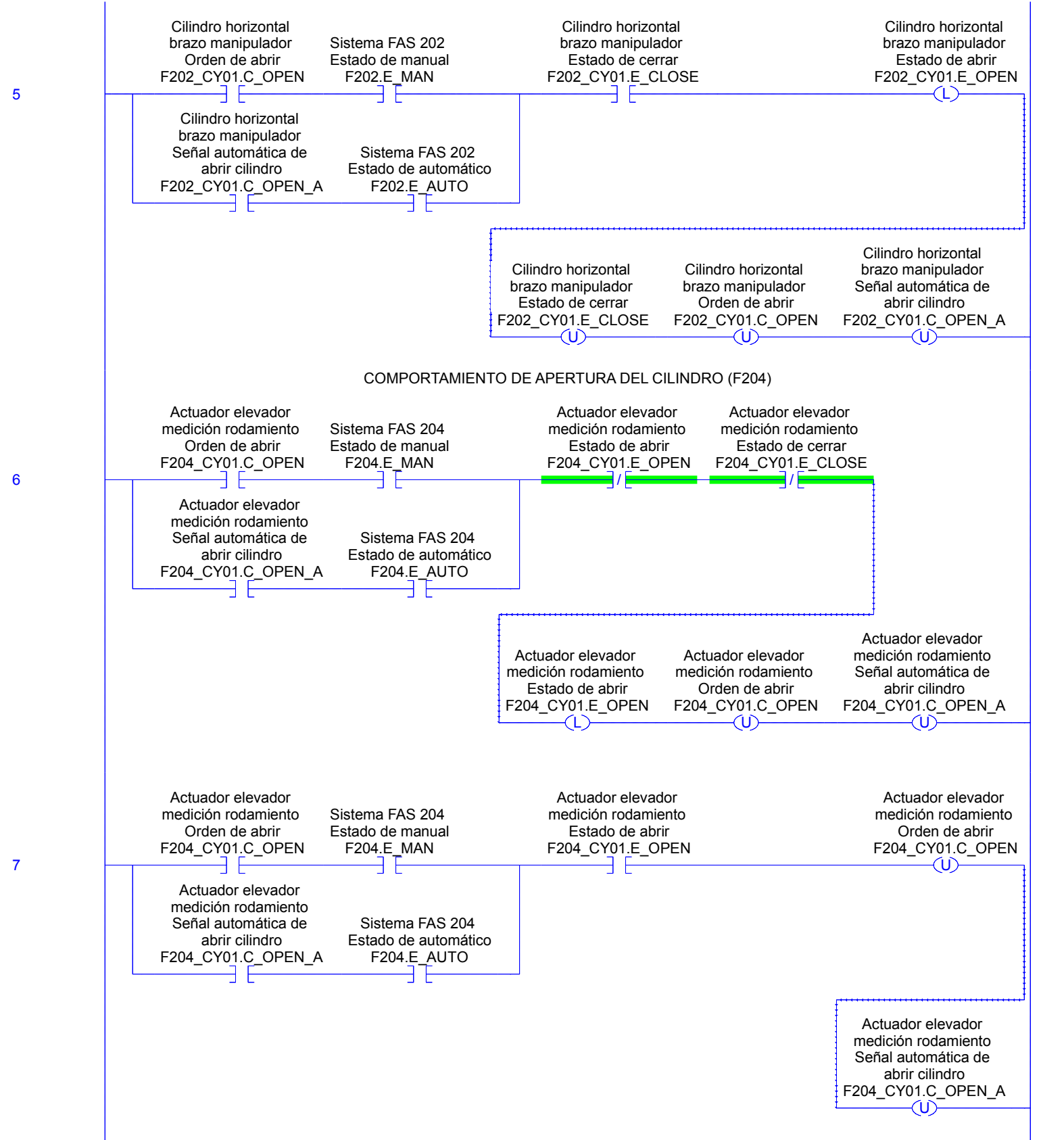


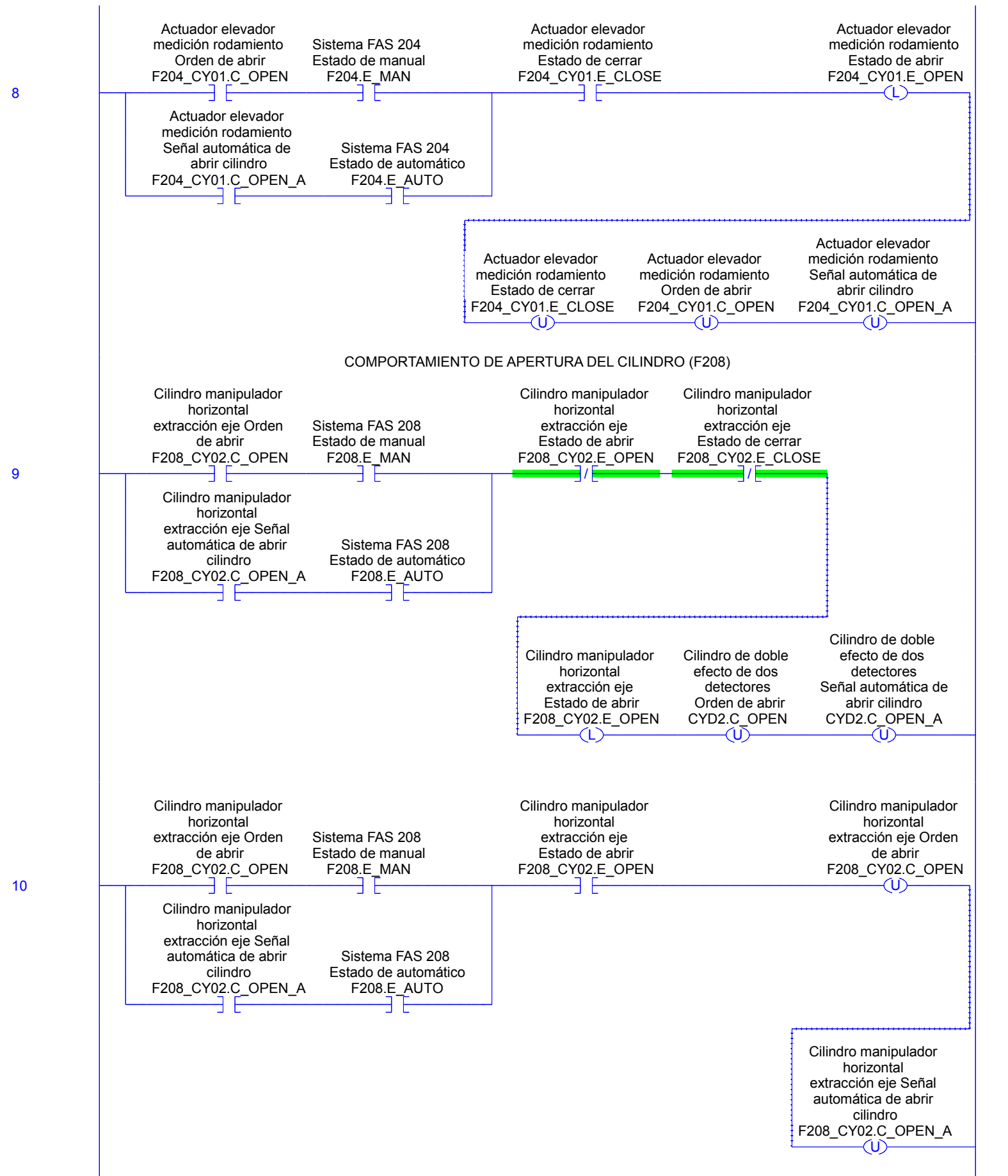


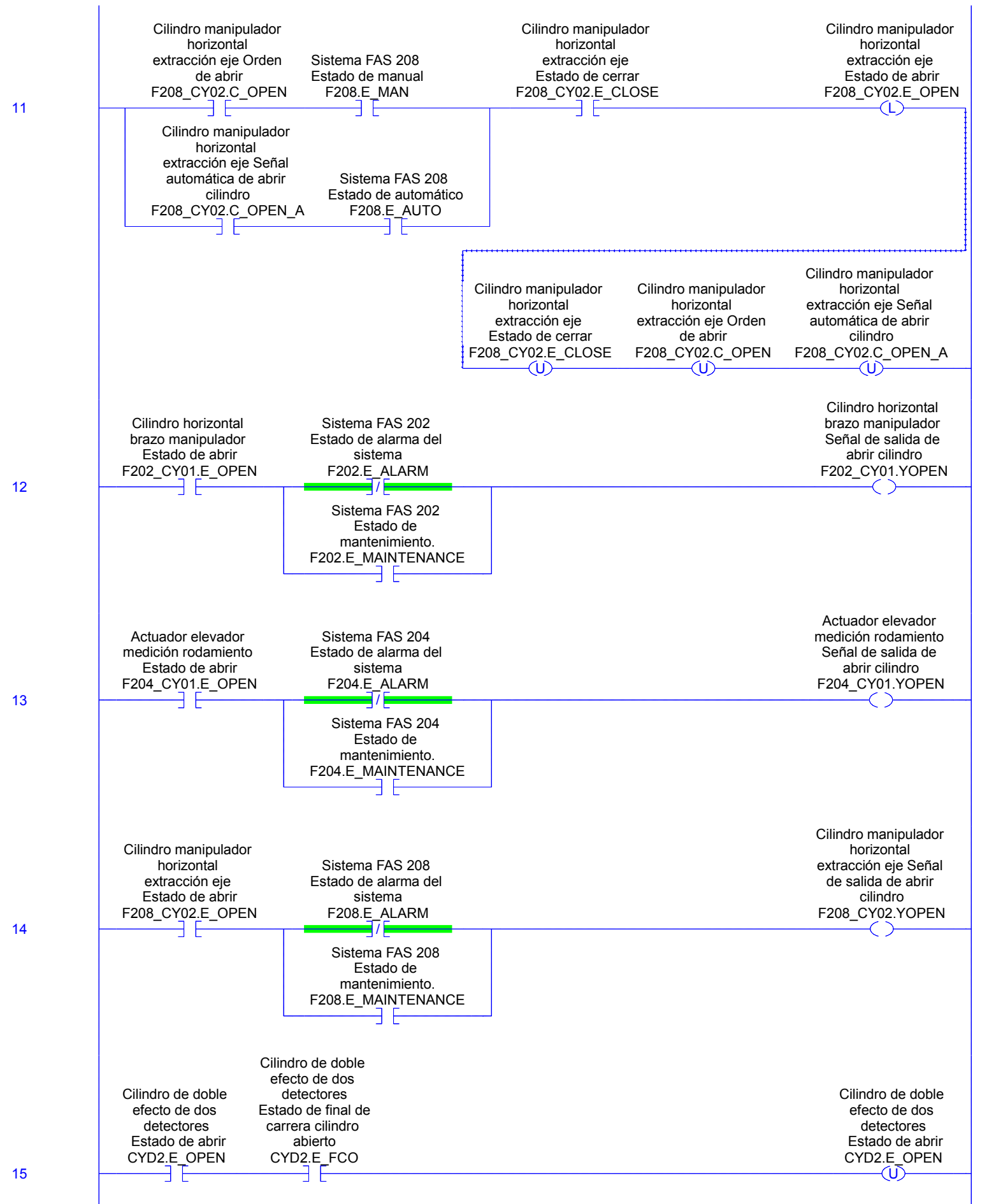












COMPORTAMIENTO DE CIERRE DEL CILINDRO (F202)

16

Cilindro horizontal
brazo manipulador
Orden de cerrar
F202_CY01.C_CLOSE

Sistema FAS 202
Estado de manual
F202.E_MAN

Cilindro horizontal
brazo manipulador
Estado de abrir
F202_CY01.E_OPEN

Cilindro horizontal
brazo manipulador
Estado de cerrar
F202_CY01.E_CLOSE

Cilindro horizontal
brazo manipulador
Señal automática de
cerrar cilindro
F202_CY01.C_CLOSE_A

Sistema FAS 202
Estado de automático
F202.E_AUTO

Cilindro horizontal
brazo manipulador
Estado de cerrar
F202_CY01.E_CLOSE

Cilindro horizontal
brazo manipulador
Orden de cerrar
F202_CY01.C_CLOSE

Cilindro horizontal
brazo manipulador
Señal automática de
cerrar cilindro
F202_CY01.C_CLOSE_A

17

Cilindro horizontal
brazo manipulador
Orden de cerrar
F202_CY01.C_CLOSE

Sistema FAS 202
Estado de manual
F202.E_MAN

Cilindro horizontal
brazo manipulador
Estado de cerrar
F202_CY01.E_CLOSE

Cilindro horizontal
brazo manipulador
Orden de cerrar
F202_CY01.C_CLOSE

Cilindro horizontal
brazo manipulador
Señal automática de
cerrar cilindro
F202_CY01.C_CLOSE_A

Sistema FAS 202
Estado de automático
F202.E_AUTO

Cilindro horizontal
brazo manipulador
Señal automática de
cerrar cilindro
F202_CY01.C_CLOSE_A

18

Cilindro horizontal
brazo manipulador
Orden de cerrar
F202_CY01.C_CLOSE

Sistema FAS 202
Estado de manual
F202.E_MAN

Cilindro horizontal
brazo manipulador
Estado de abrir
F202_CY01.E_OPEN

Cilindro horizontal
brazo manipulador
Estado de cerrar
F202_CY01.E_CLOSE

Cilindro horizontal
brazo manipulador
Señal automática de
cerrar cilindro
F202_CY01.C_CLOSE_A

Sistema FAS 202
Estado de automático
F202.E_AUTO

Cilindro horizontal
brazo manipulador
Estado de abrir
F202_CY01.E_OPEN

Cilindro horizontal
brazo manipulador
Orden de cerrar
F202_CY01.C_CLOSE

Cilindro horizontal
brazo manipulador
Señal automática de
cerrar cilindro
F202_CY01.C_CLOSE_A

COMPORTAMIENTO DE CIERRE DEL CILINDRO (F204)

19

Actuador elevador
medición rodamiento
Orden de cerrar
F204_CY01.C_CLOSE

Sistema FAS 204
Estado de manual
F204.E_MAN

Actuador elevador
medición rodamiento
Estado de abrir
F204_CY01.E_OPEN

Actuador elevador
medición rodamiento
Estado de cerrar
F204_CY01.E_CLOSE

Actuador elevador
medición rodamiento
Señal automática de
cerrar cilindro
F204_CY01.C_CLOSE_A

Sistema FAS 204
Estado de automático
F204.E_AUTO

Actuador elevador
medición rodamiento
Estado de cerrar
F204_CY01.E_CLOSE

Actuador elevador
medición rodamiento
Orden de cerrar
F204_CY01.C_CLOSE

Actuador elevador
medición rodamiento
Señal automática de
cerrar cilindro
F204_CY01.C_CLOSE_A

20

Actuador elevador
medición rodamiento
Orden de cerrar
F204_CY01.C_CLOSE

Sistema FAS 204
Estado de manual
F204.E_MAN

Actuador elevador
medición rodamiento
Estado de cerrar
F204_CY01.E_CLOSE

Actuador elevador
medición rodamiento
Orden de cerrar
F204_CY01.C_CLOSE

Actuador elevador
medición rodamiento
Señal automática de
cerrar cilindro
F204_CY01.C_CLOSE_A

Sistema FAS 204
Estado de automático
F204.E_AUTO

Actuador elevador
medición rodamiento
Señal automática de
cerrar cilindro
F204_CY01.C_CLOSE_A

21

Actuador elevador
medición rodamiento
Orden de cerrar
F204_CY01.C_CLOSE

Sistema FAS 204
Estado de manual
F204.E_MAN

Actuador elevador
medición rodamiento
Estado de abrir
F204_CY01.E_OPEN

Actuador elevador
medición rodamiento
Estado de cerrar
F204_CY01.E_CLOSE

Actuador elevador
medición rodamiento
Señal automática de
cerrar cilindro
F204_CY01.C_CLOSE_A

Sistema FAS 204
Estado de automático
F204.E_AUTO

Actuador elevador
medición rodamiento
Estado de abrir
F204_CY01.E_OPEN

Actuador elevador
medición rodamiento
Orden de cerrar
F204_CY01.C_CLOSE

Actuador elevador
medición rodamiento
Señal automática de
cerrar cilindro
F204_CY01.C_CLOSE_A

COMPORTAMIENTO DE CIERRE DEL CILINDRO (F208)

22

Cilindro manipulador
horizontal
extracción eje Orden
de cerrar
F208_CY02.C_CLOSE

Sistema FAS 208
Estado de manual
F208.E_MAN

Cilindro manipulador
horizontal
extracción eje
Estado de abrir
F208_CY02.E_OPEN

Cilindro manipulador
horizontal
extracción eje
Estado de cerrar
F208_CY02.E_CLOSE

Cilindro manipulador
horizontal
extracción eje Señal
automática de cerrar
cilindro
F208_CY02.C_CLOSE_A

Sistema FAS 208
Estado de automático
F208.E_AUTO

Cilindro manipulador
horizontal
extracción eje
Estado de cerrar
F208_CY02.E_CLOSE

Cilindro manipulador
horizontal
extracción eje Orden
de cerrar
F208_CY02.C_CLOSE

Cilindro manipulador
horizontal
extracción eje Señal
automática de cerrar
cilindro
F208_CY02.C_CLOSE_A

23

Cilindro manipulador
horizontal
extracción eje Orden
de cerrar
F208_CY02.C_CLOSE

Sistema FAS 208
Estado de manual
F208.E_MAN

Cilindro manipulador
horizontal
extracción eje
Estado de cerrar
F208_CY02.E_CLOSE

Cilindro manipulador
horizontal
extracción eje Orden
de cerrar
F208_CY02.C_CLOSE

Cilindro manipulador
horizontal
extracción eje Señal
automática de cerrar
cilindro
F208_CY02.C_CLOSE_A

Sistema FAS 208
Estado de automático
F208.E_AUTO

Cilindro manipulador
horizontal
extracción eje Señal
automática de cerrar
cilindro
F208_CY02.C_CLOSE_A

24

Cilindro manipulador
horizontal
extracción eje Orden
de cerrar
F208_CY02.C_CLOSE

Sistema FAS 208
Estado de manual
F208.E_MAN

Cilindro manipulador
horizontal
extracción eje
Estado de abrir
F208_CY02.E_OPEN

Cilindro manipulador
horizontal
extracción eje
Estado de cerrar
F208_CY02.E_CLOSE

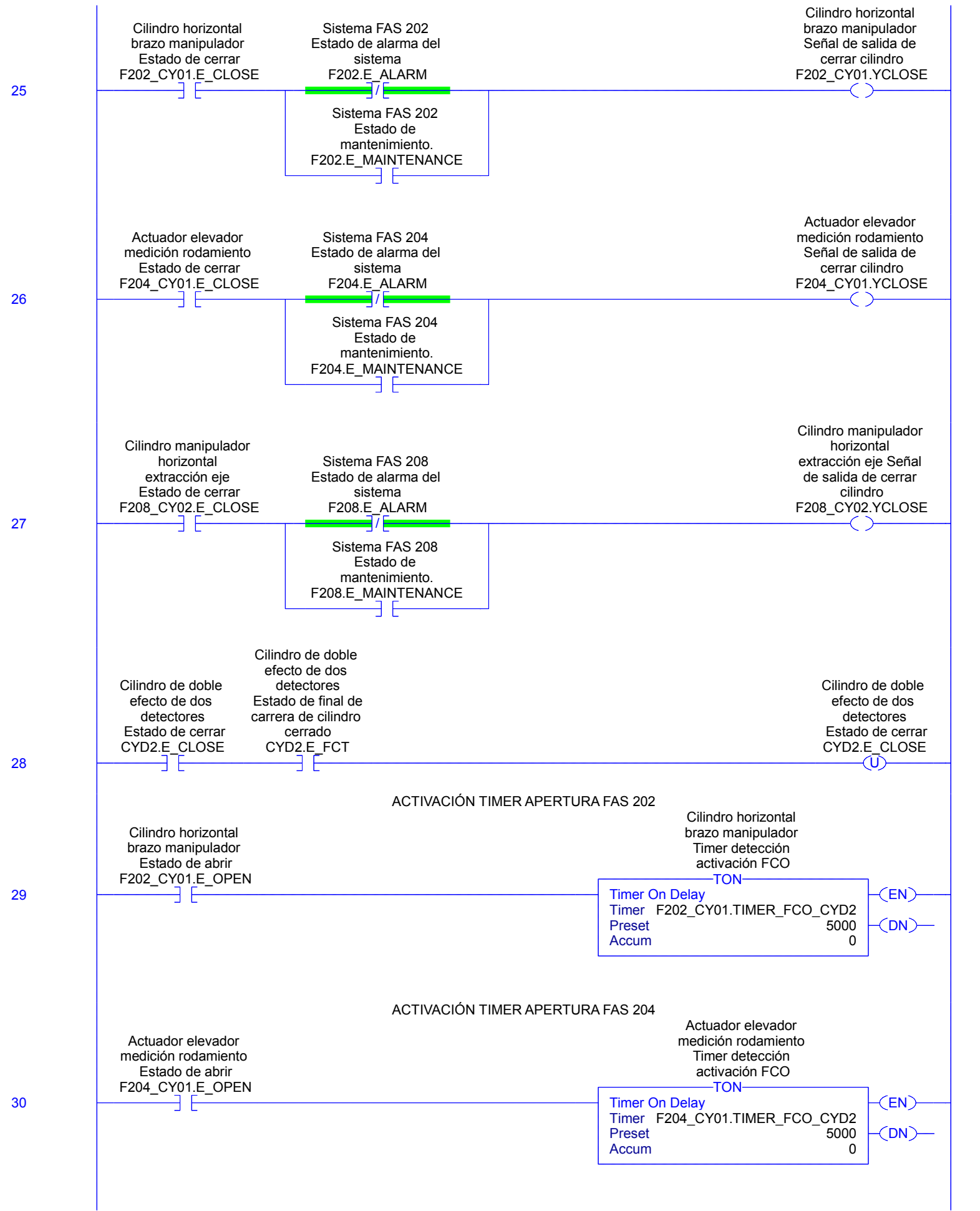
Cilindro manipulador
horizontal
extracción eje Señal
automática de cerrar
cilindro
F208_CY02.C_CLOSE_A

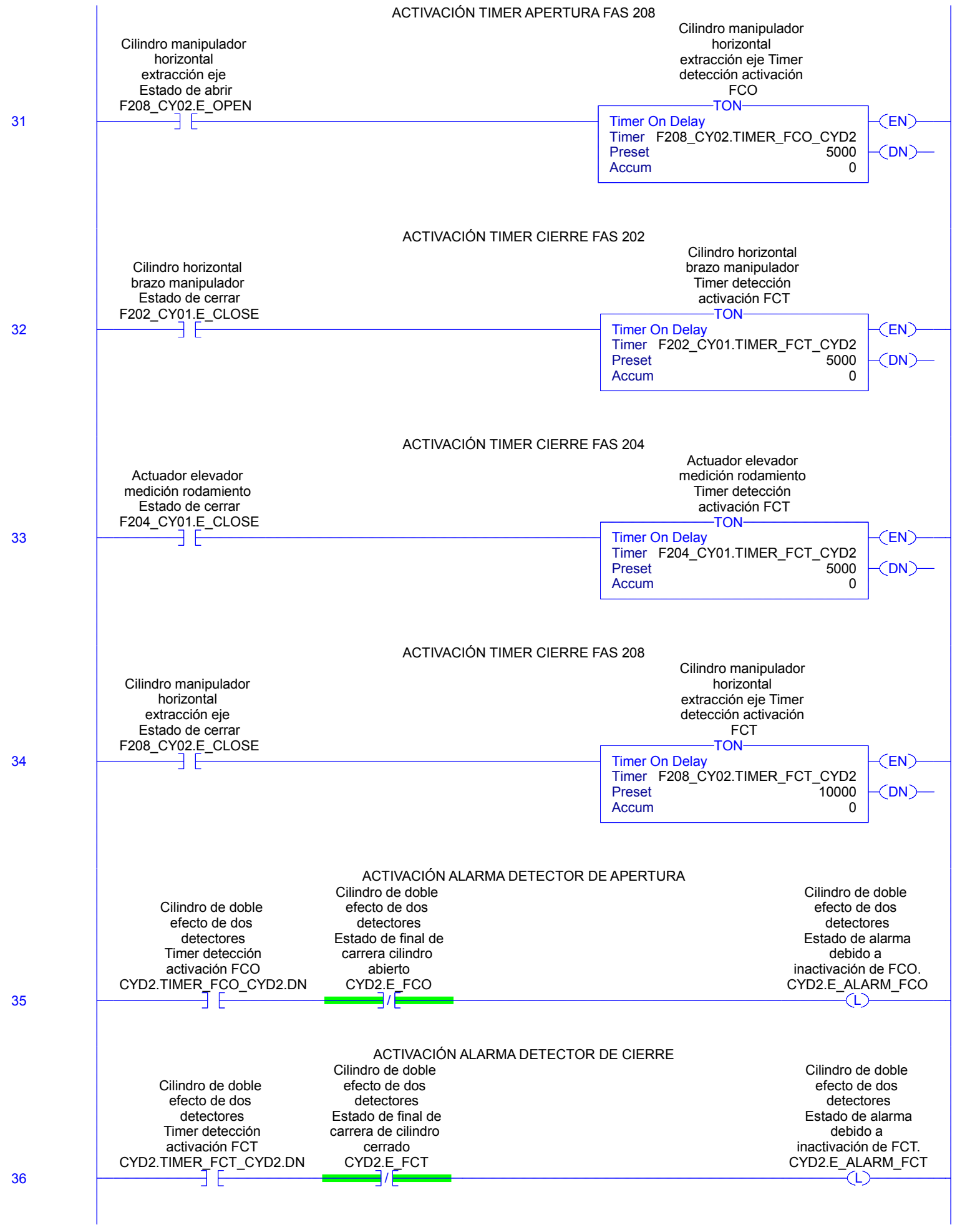
Sistema FAS 208
Estado de automático
F208.E_AUTO

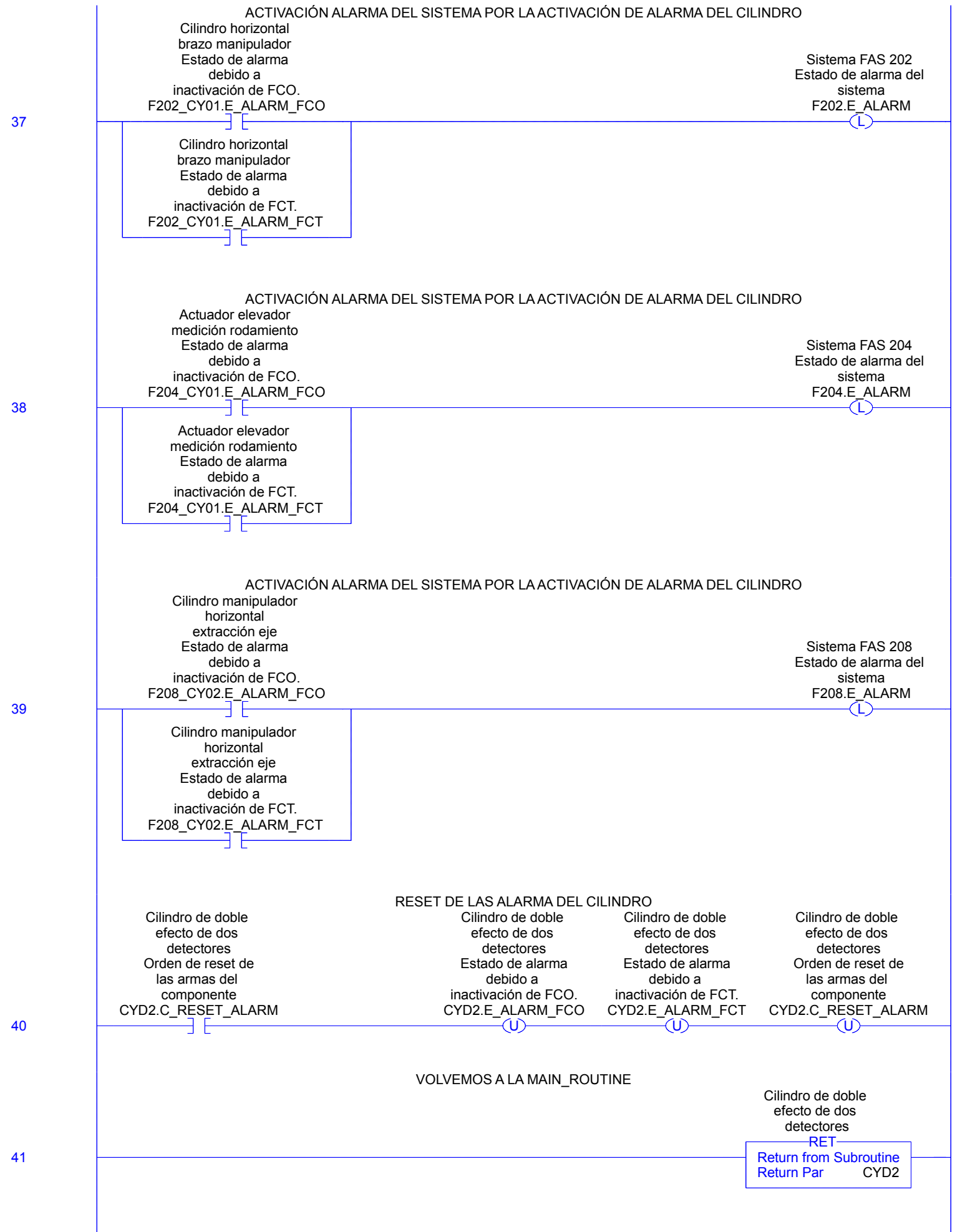
Cilindro manipulador
horizontal
extracción eje
Estado de abrir
F208_CY02.E_OPEN

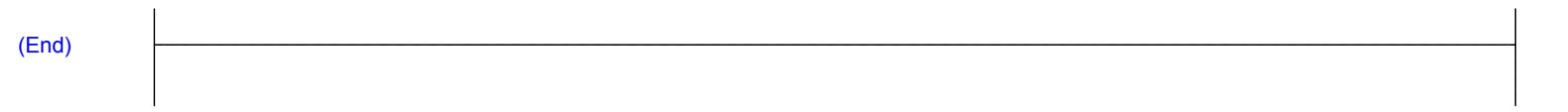
Cilindro manipulador
horizontal
extracción eje Orden
de cerrar
F208_CY02.C_CLOSE

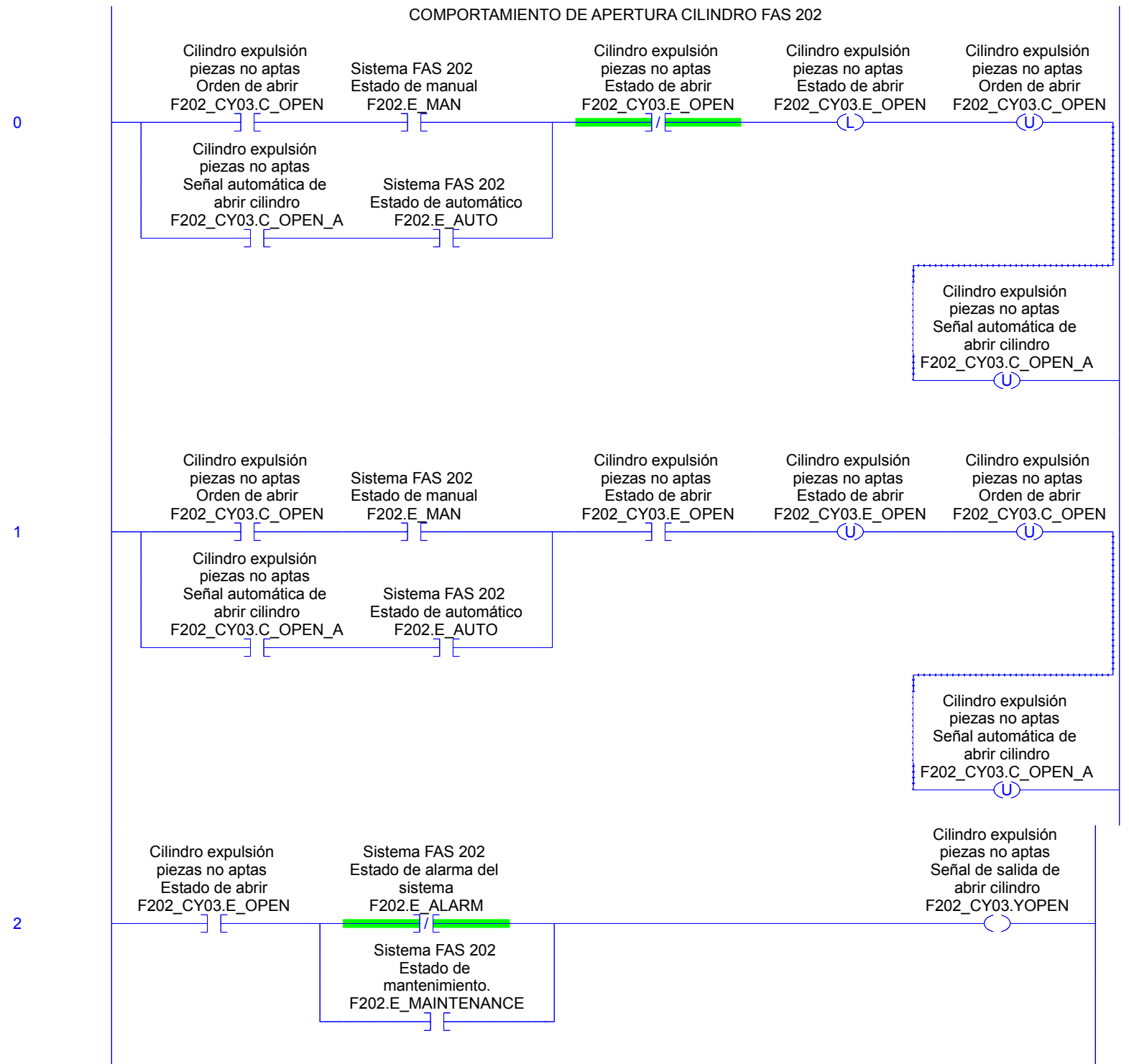
Cilindro manipulador
horizontal
extracción eje Señal
automática de cerrar
cilindro
F208_CY02.C_CLOSE_A

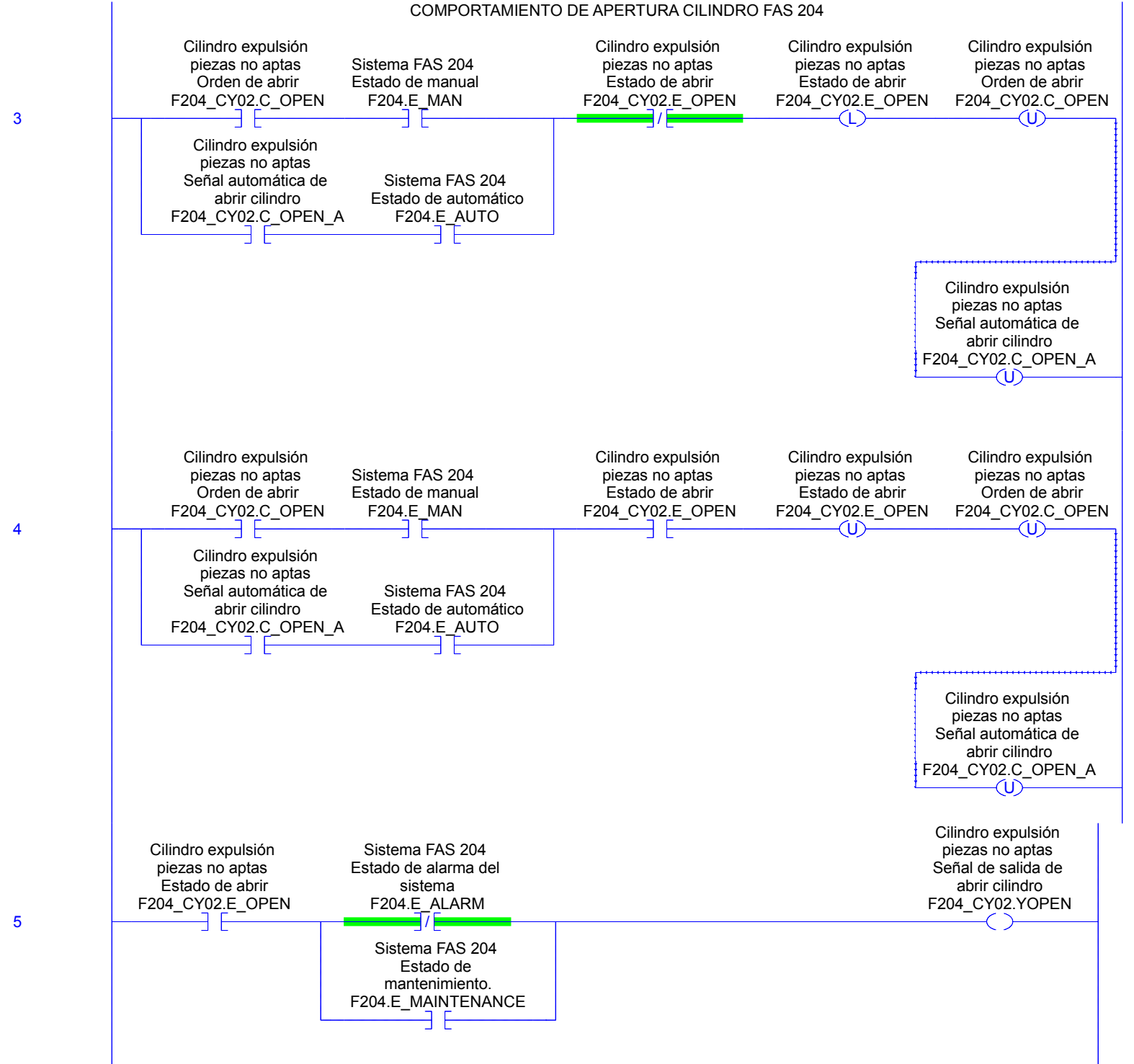


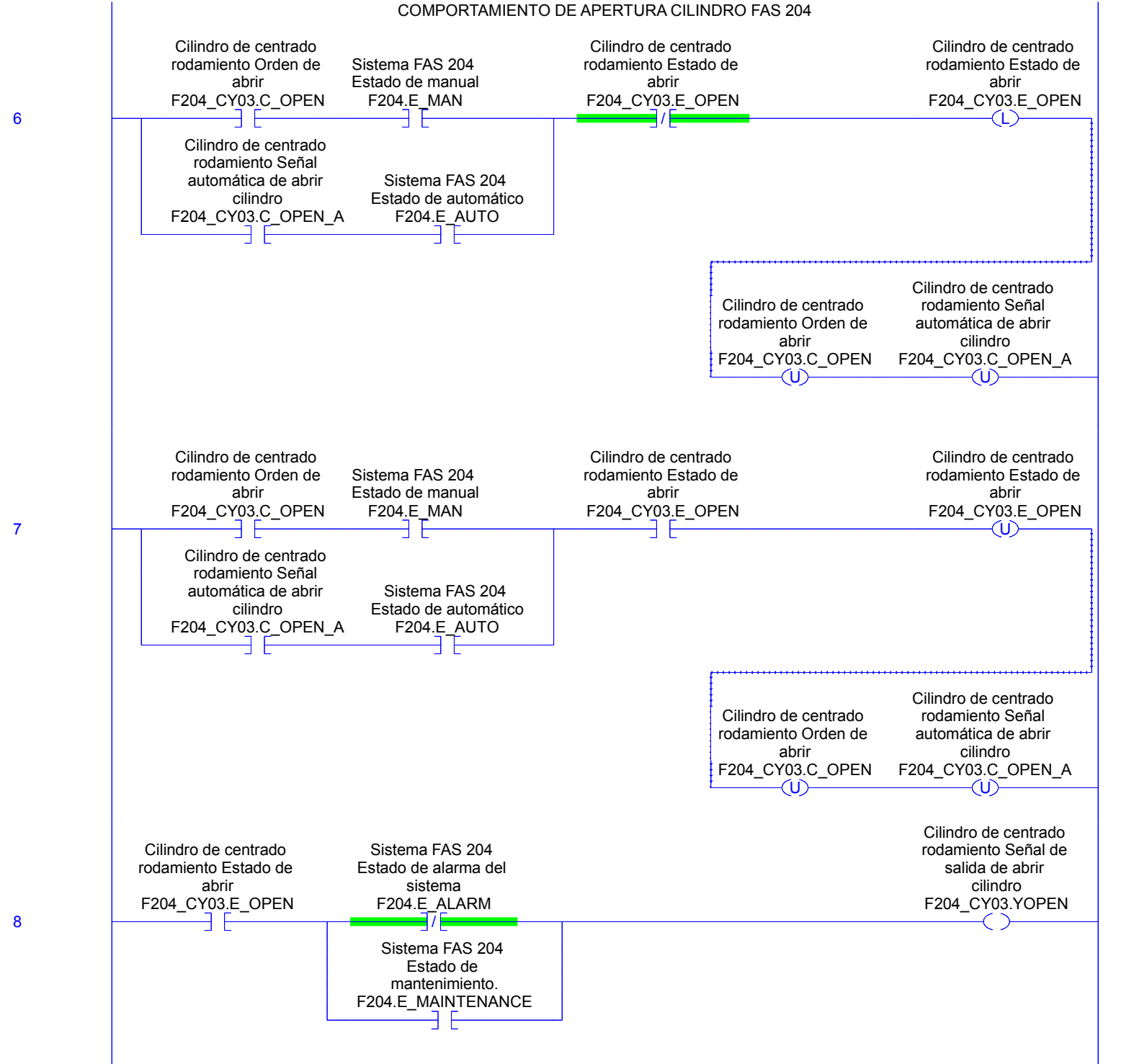


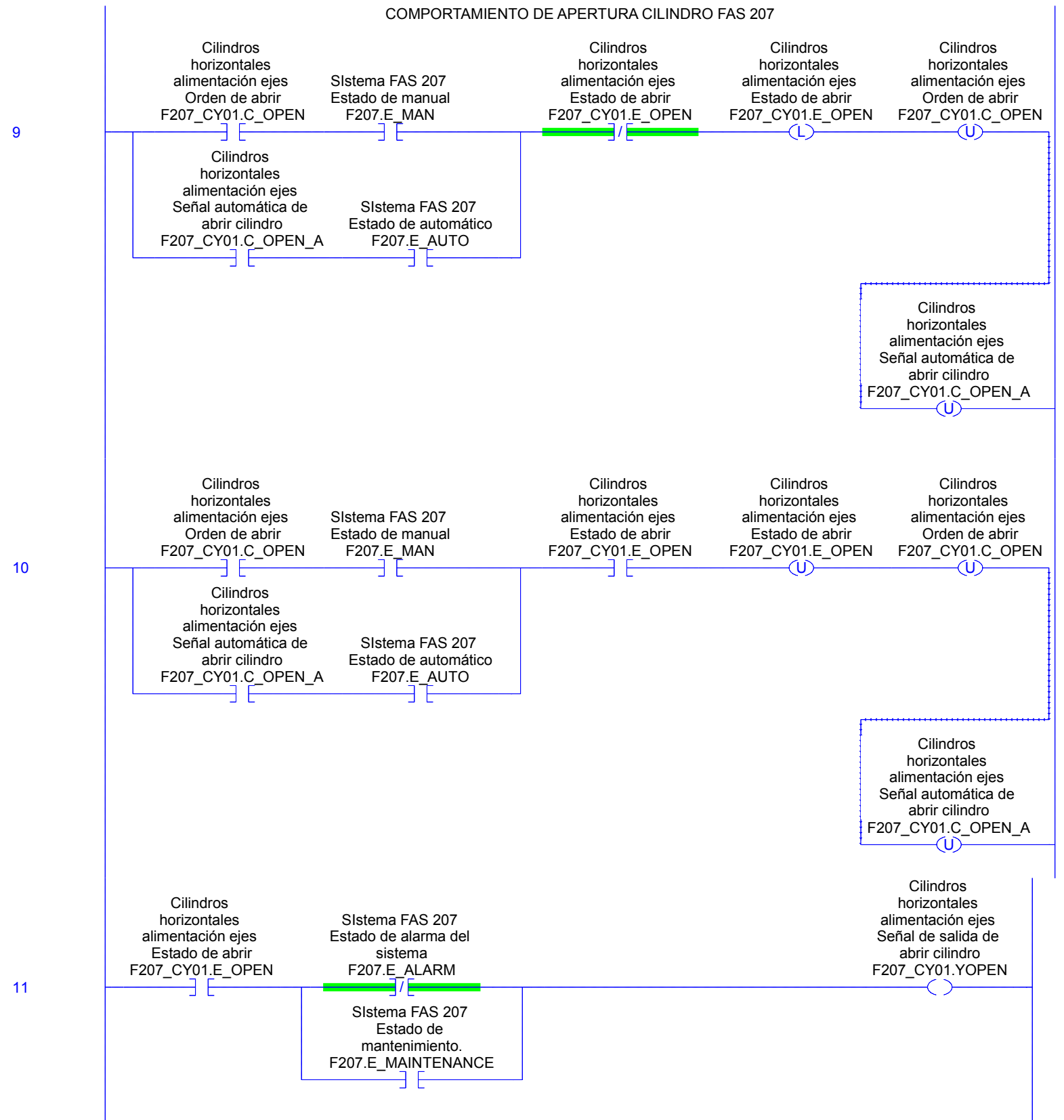


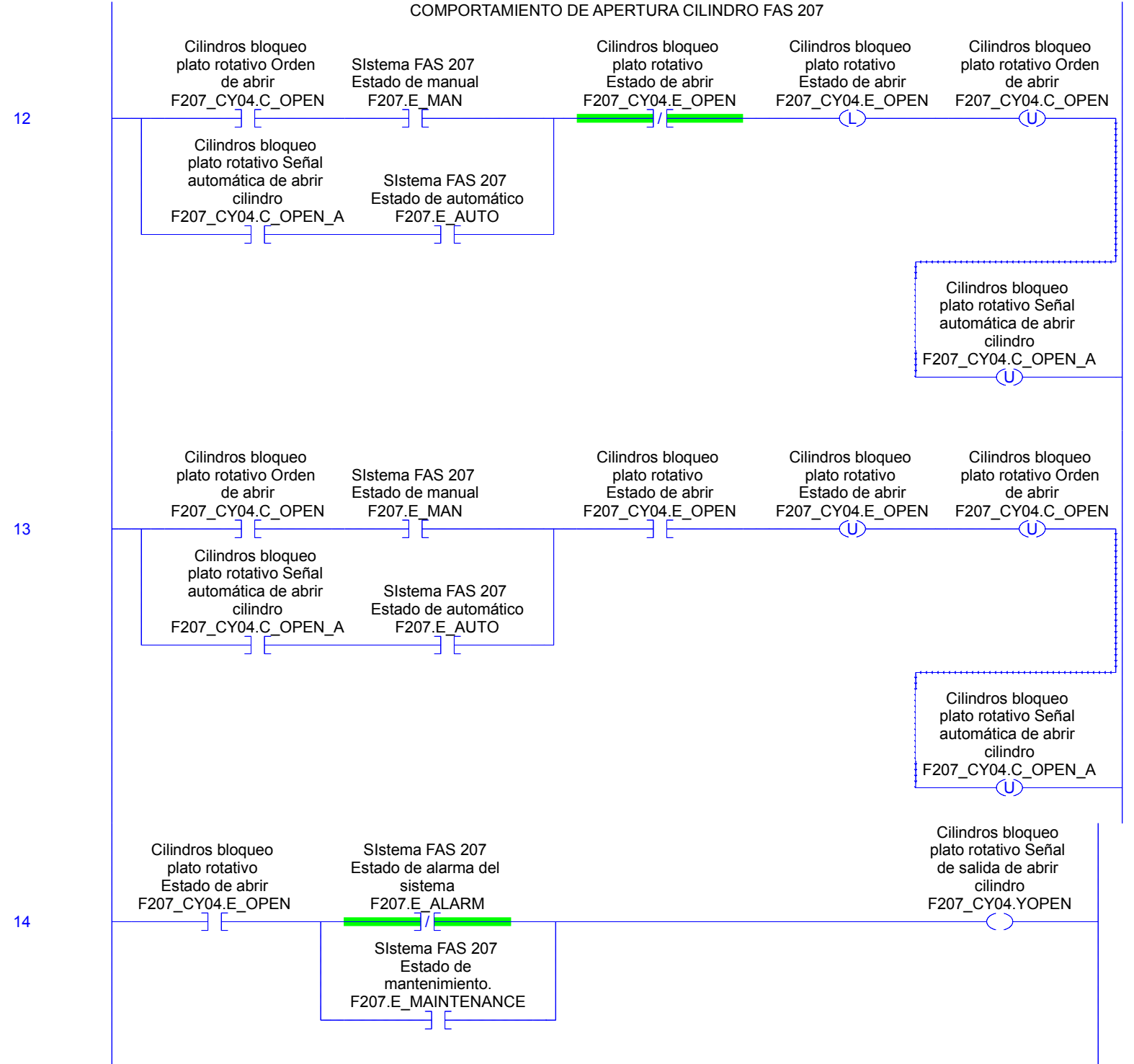




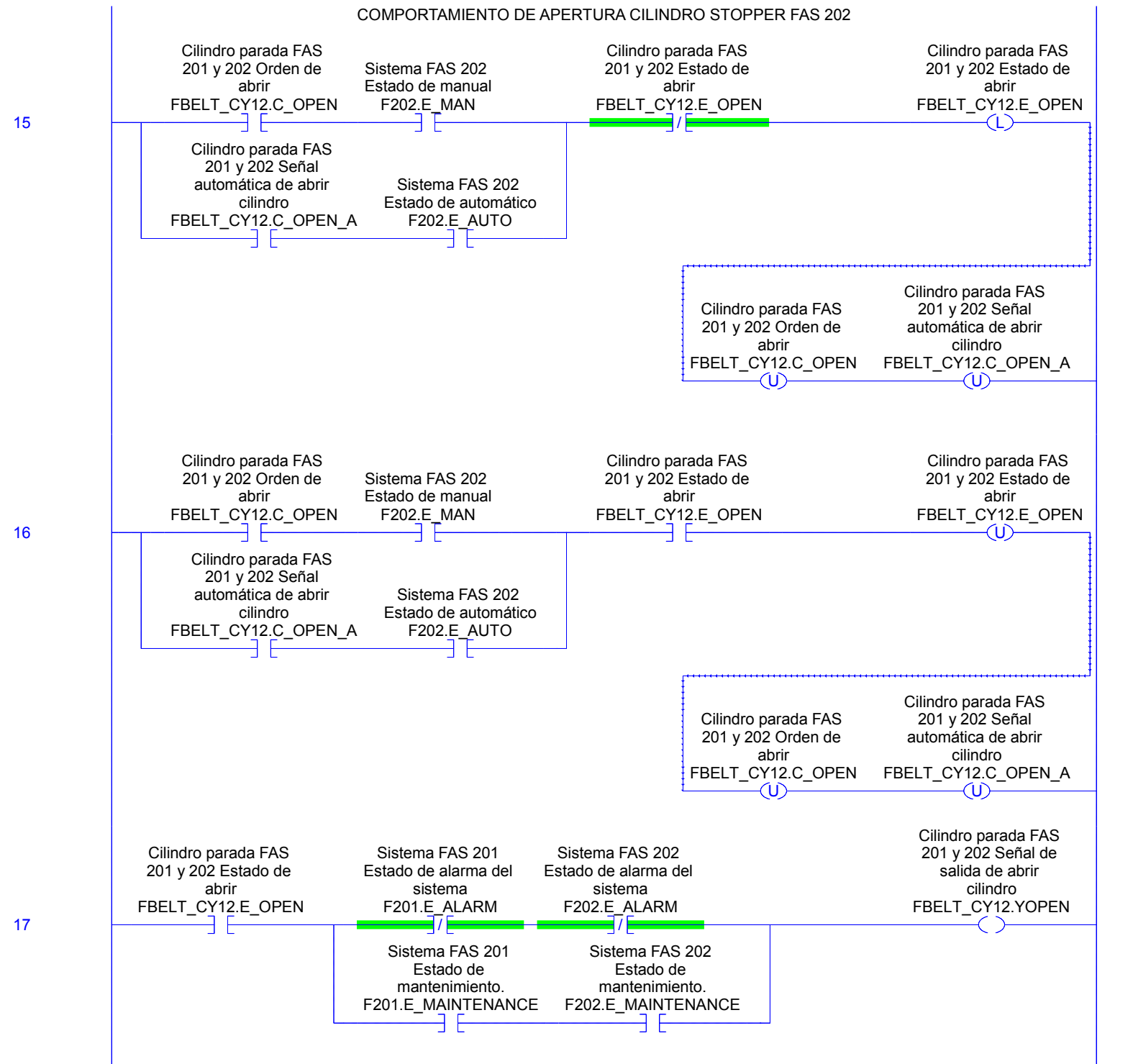


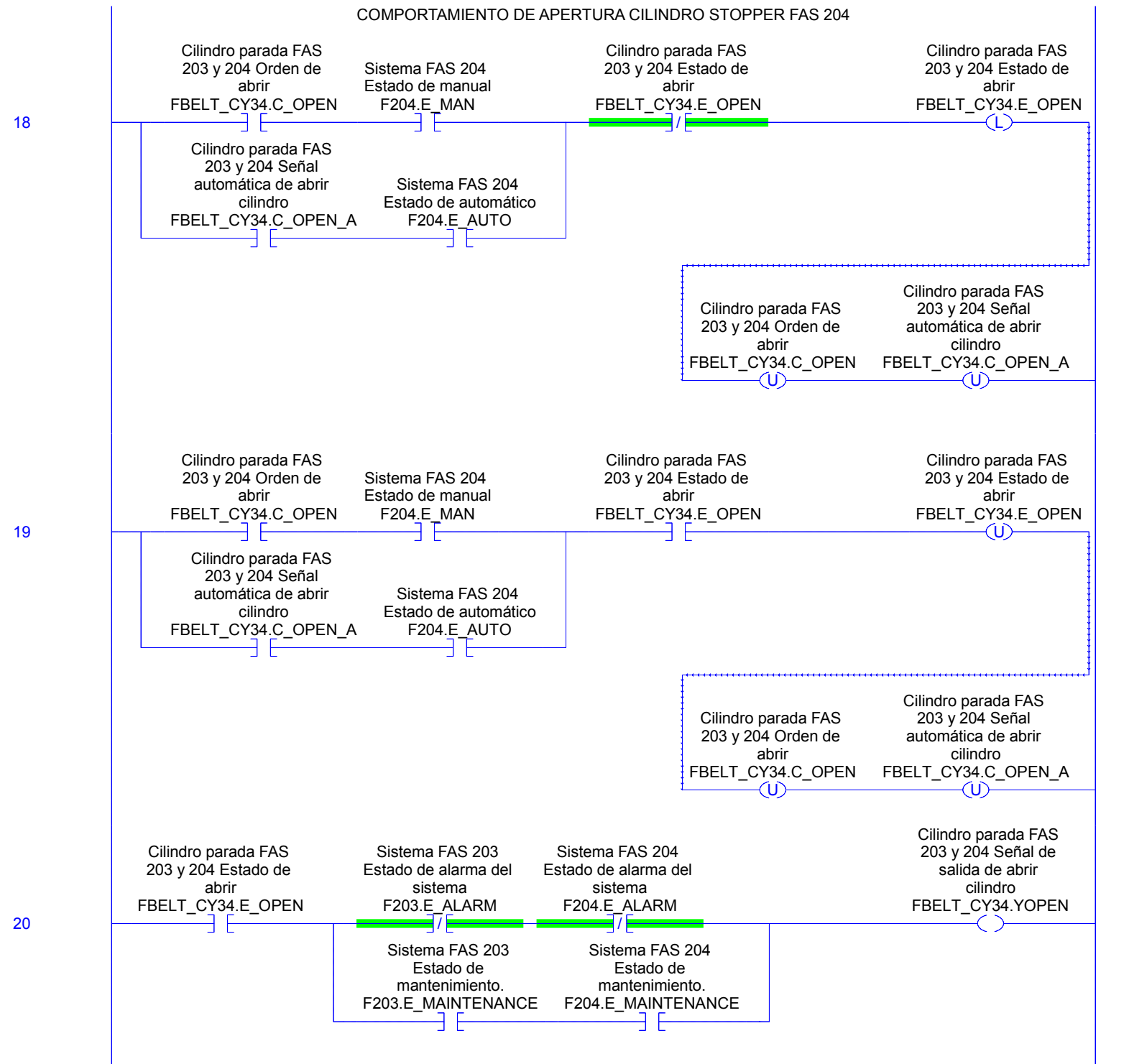


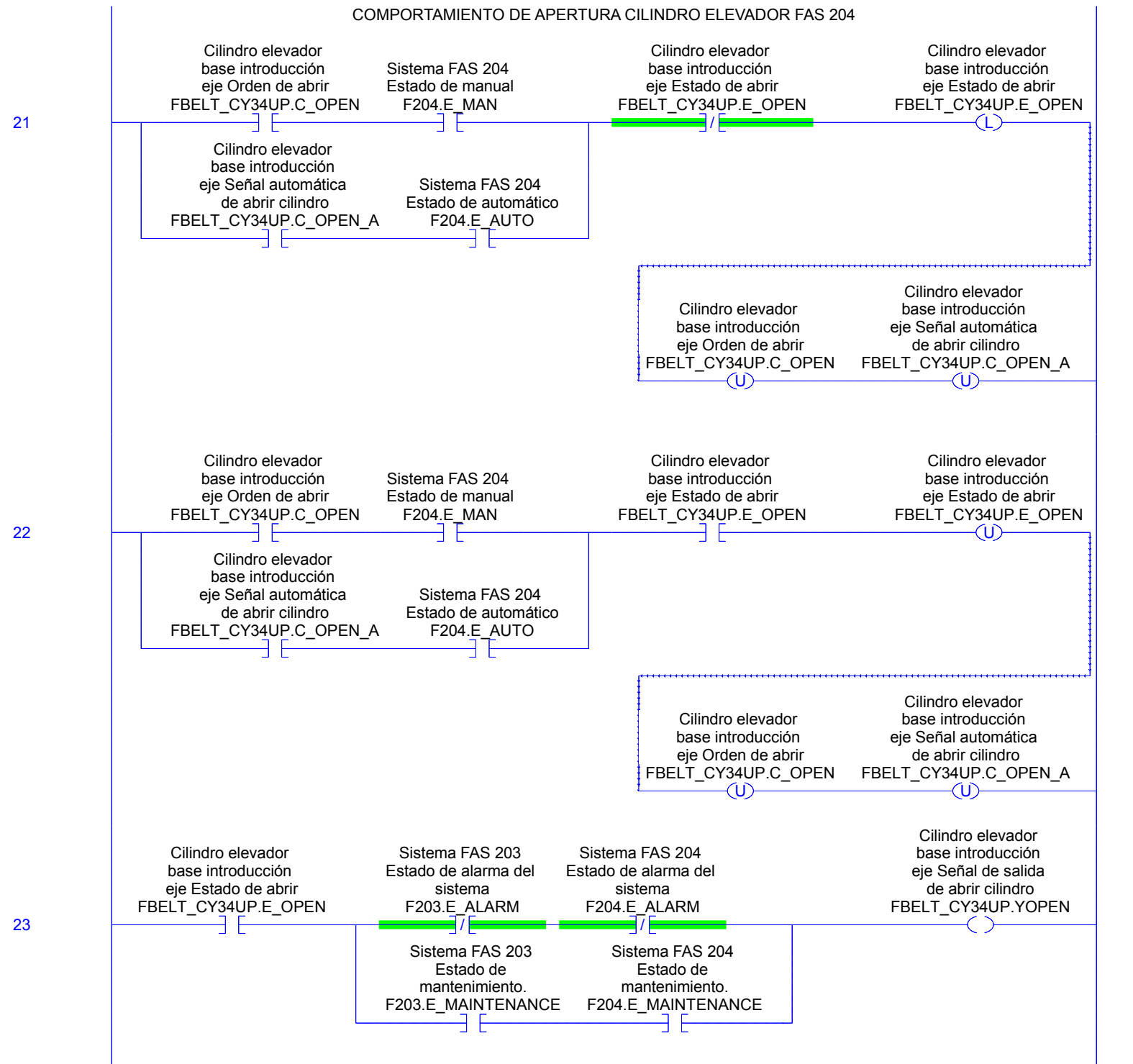


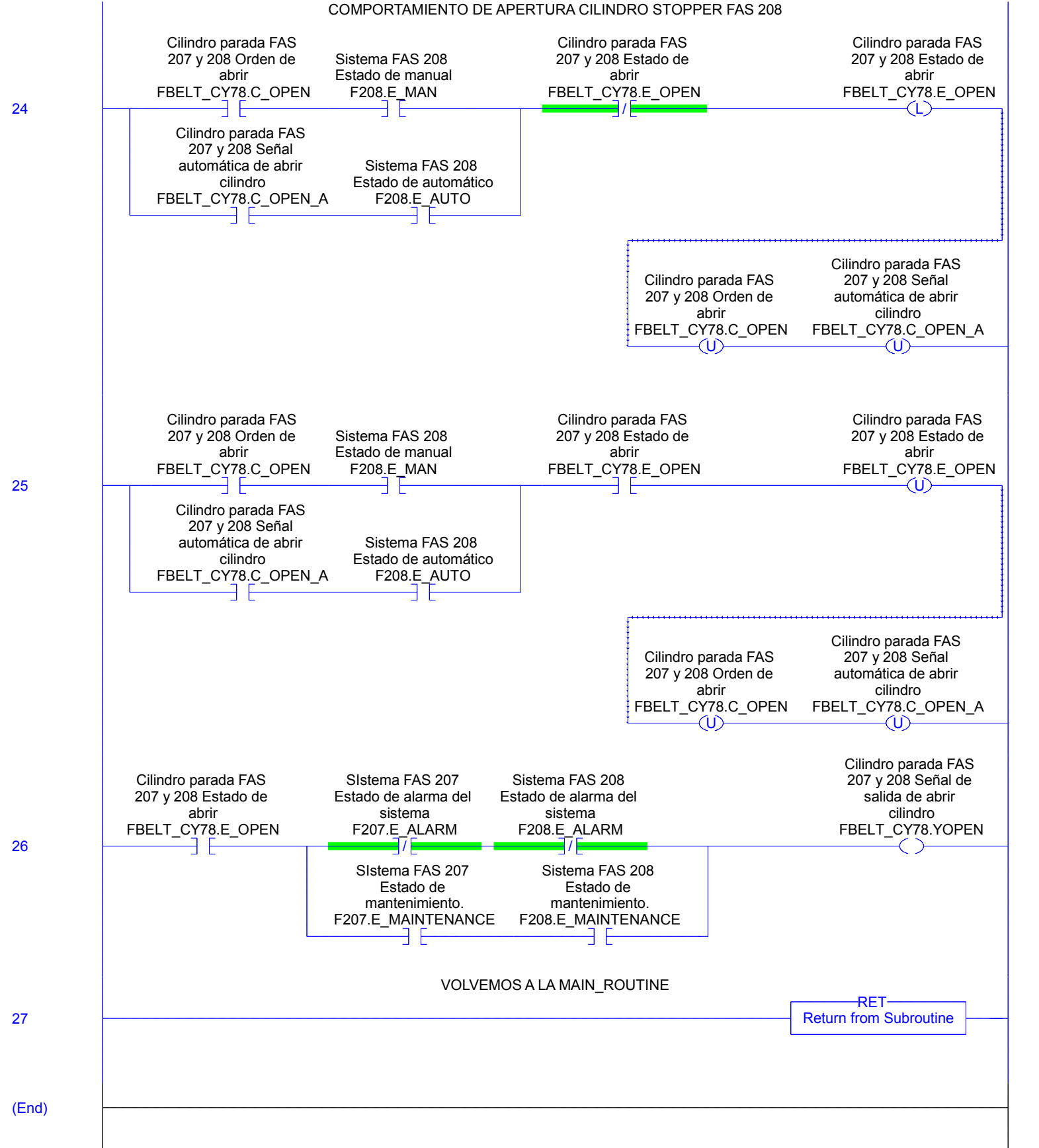


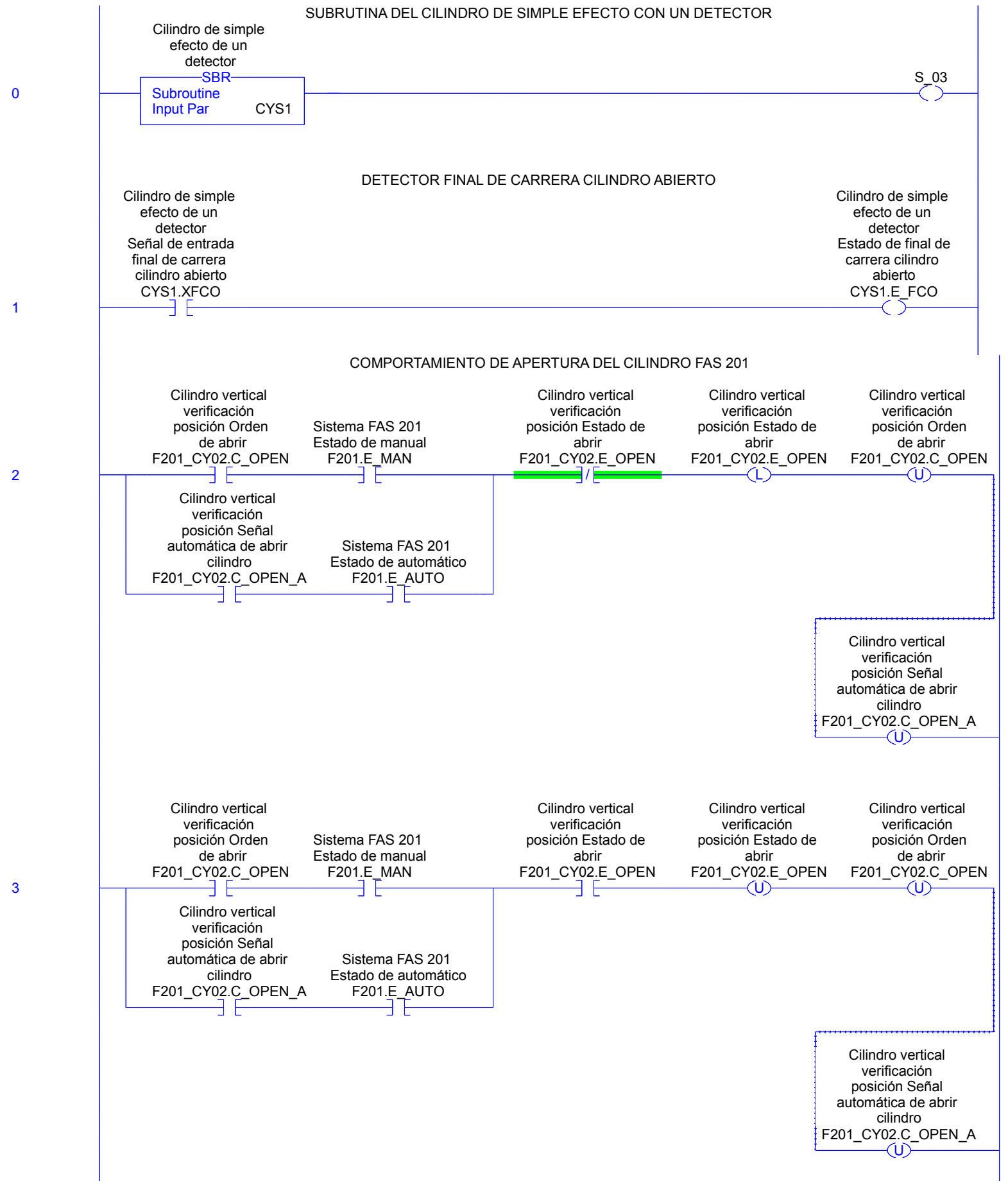
COMPORTAMIENTO DE APERTURA CILINDRO STOPPER FAS 202

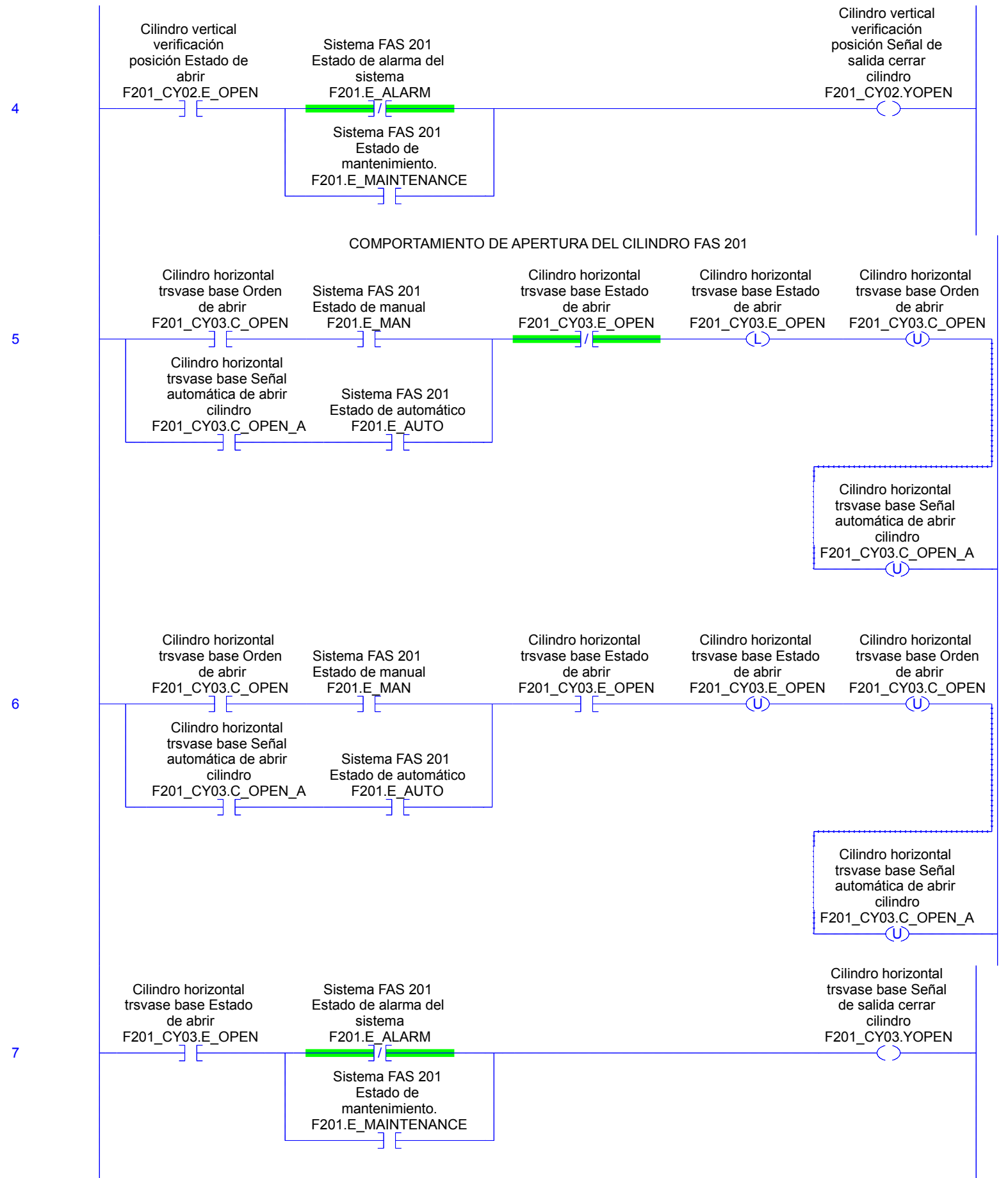


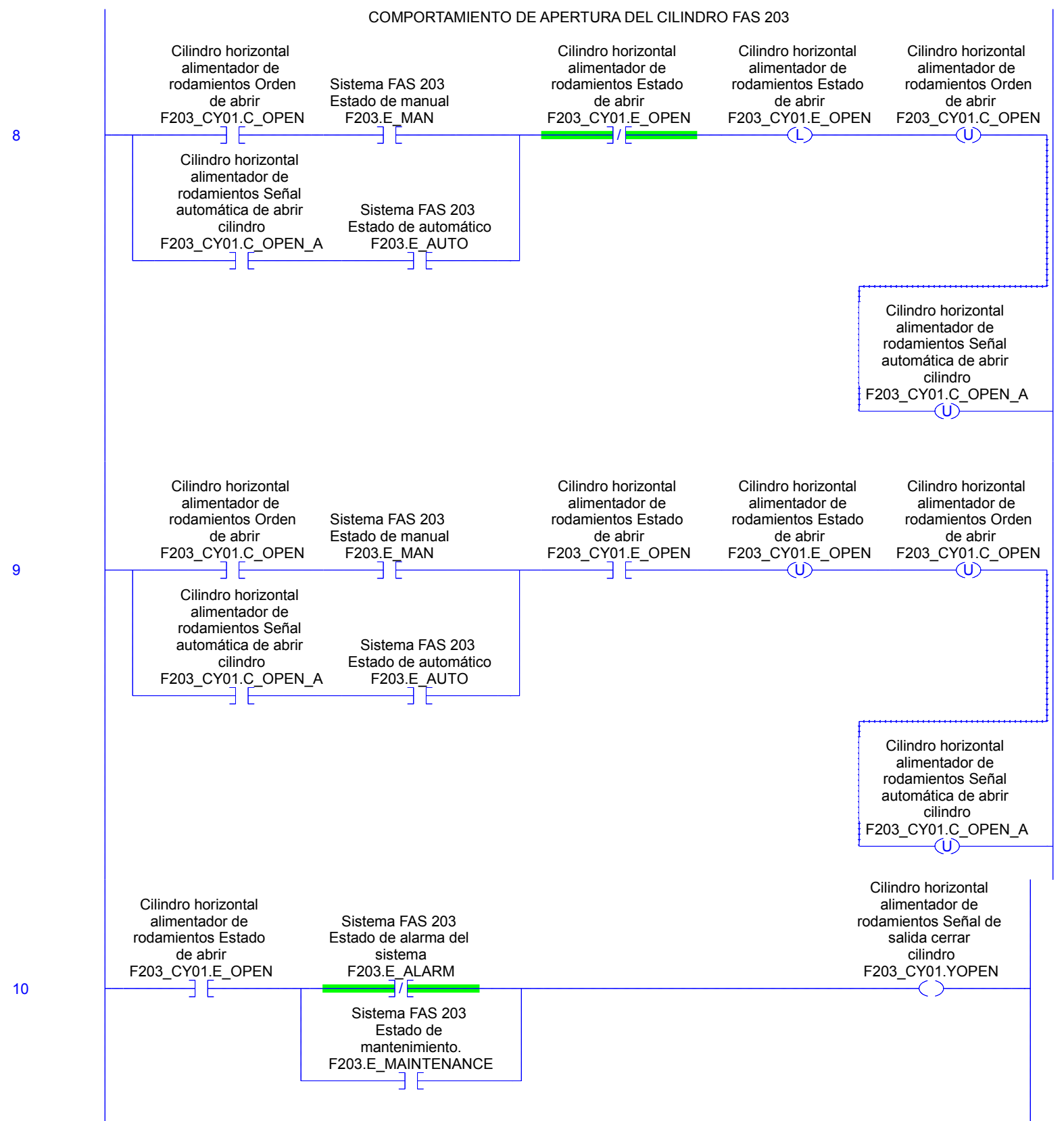


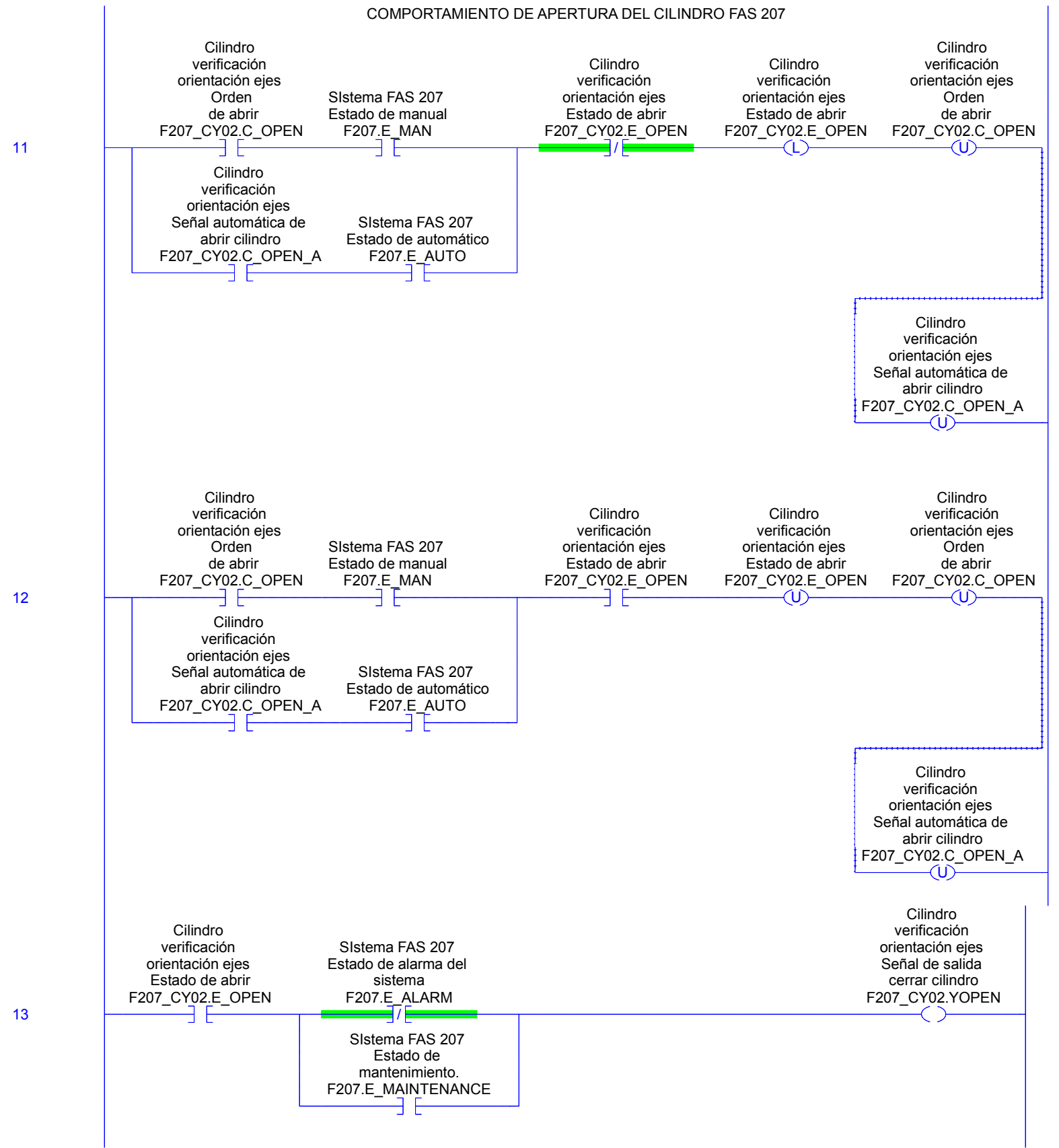


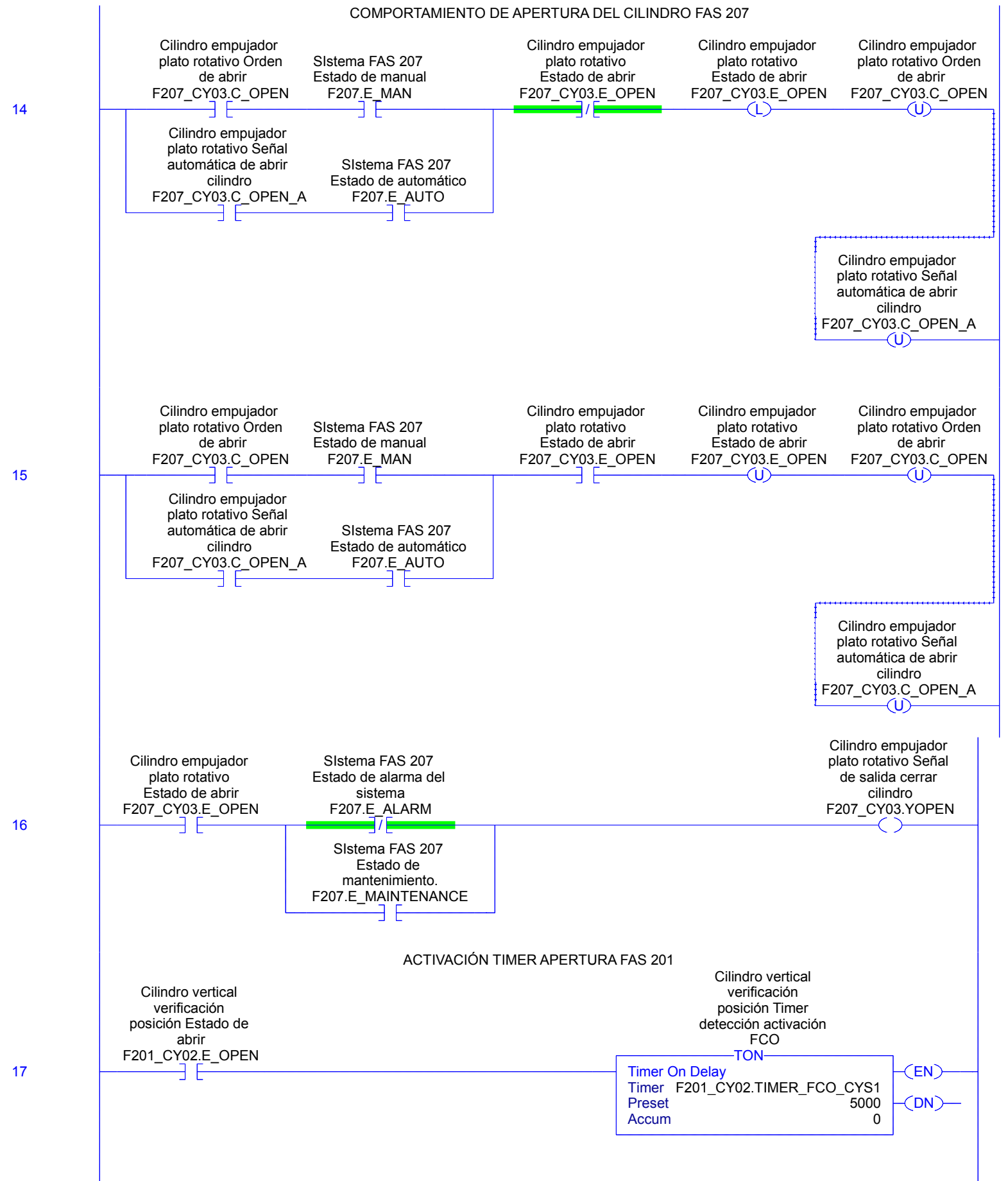


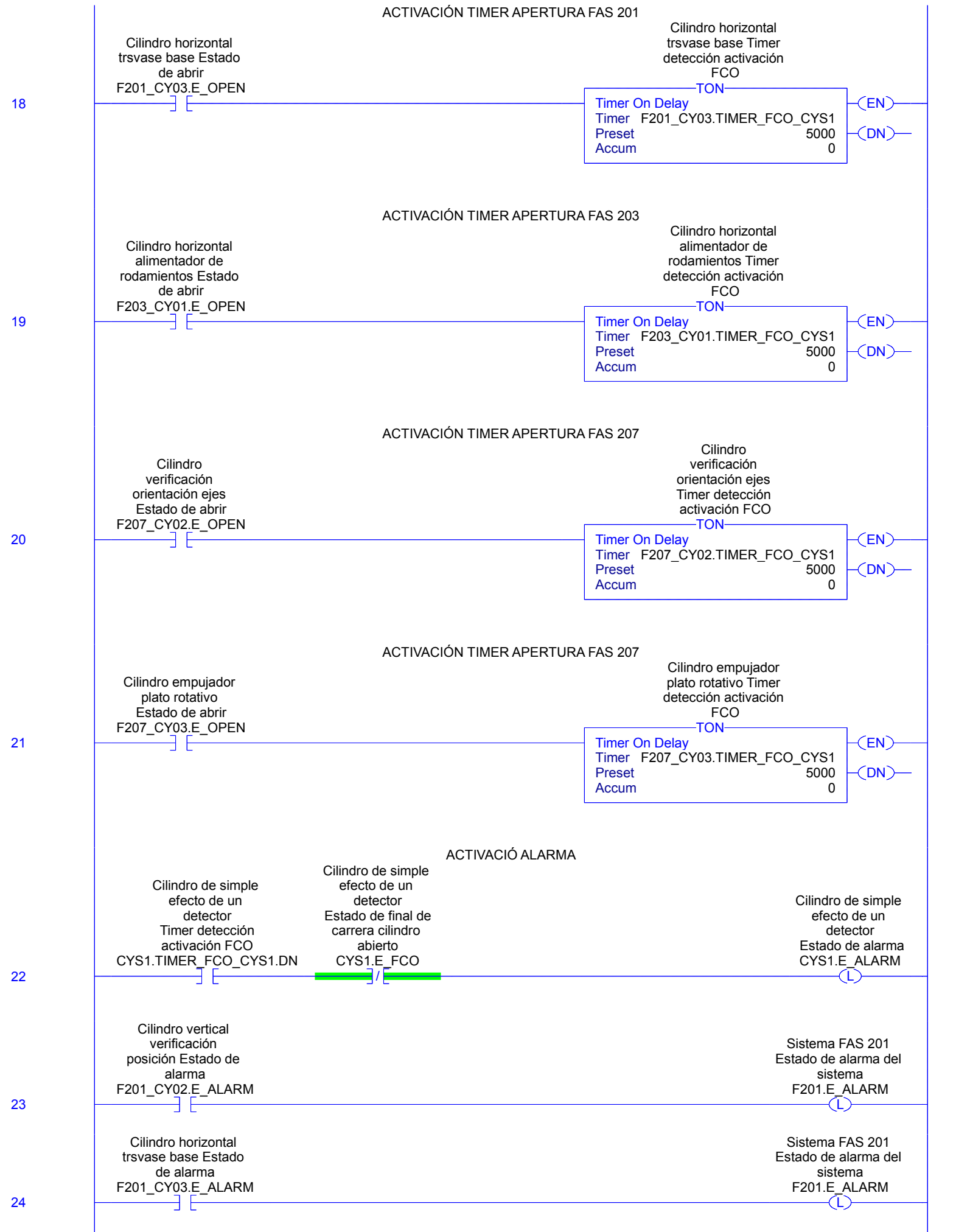


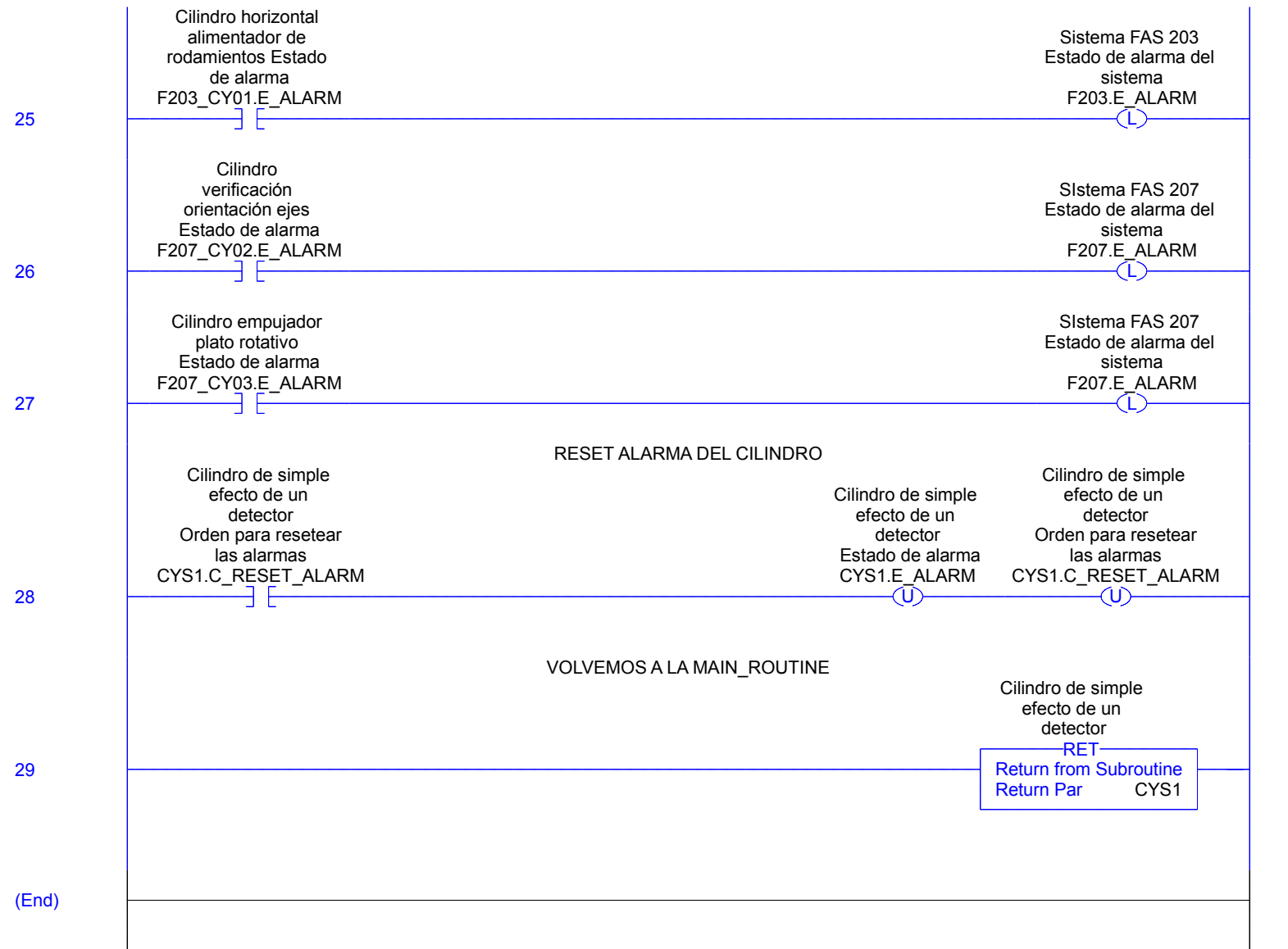


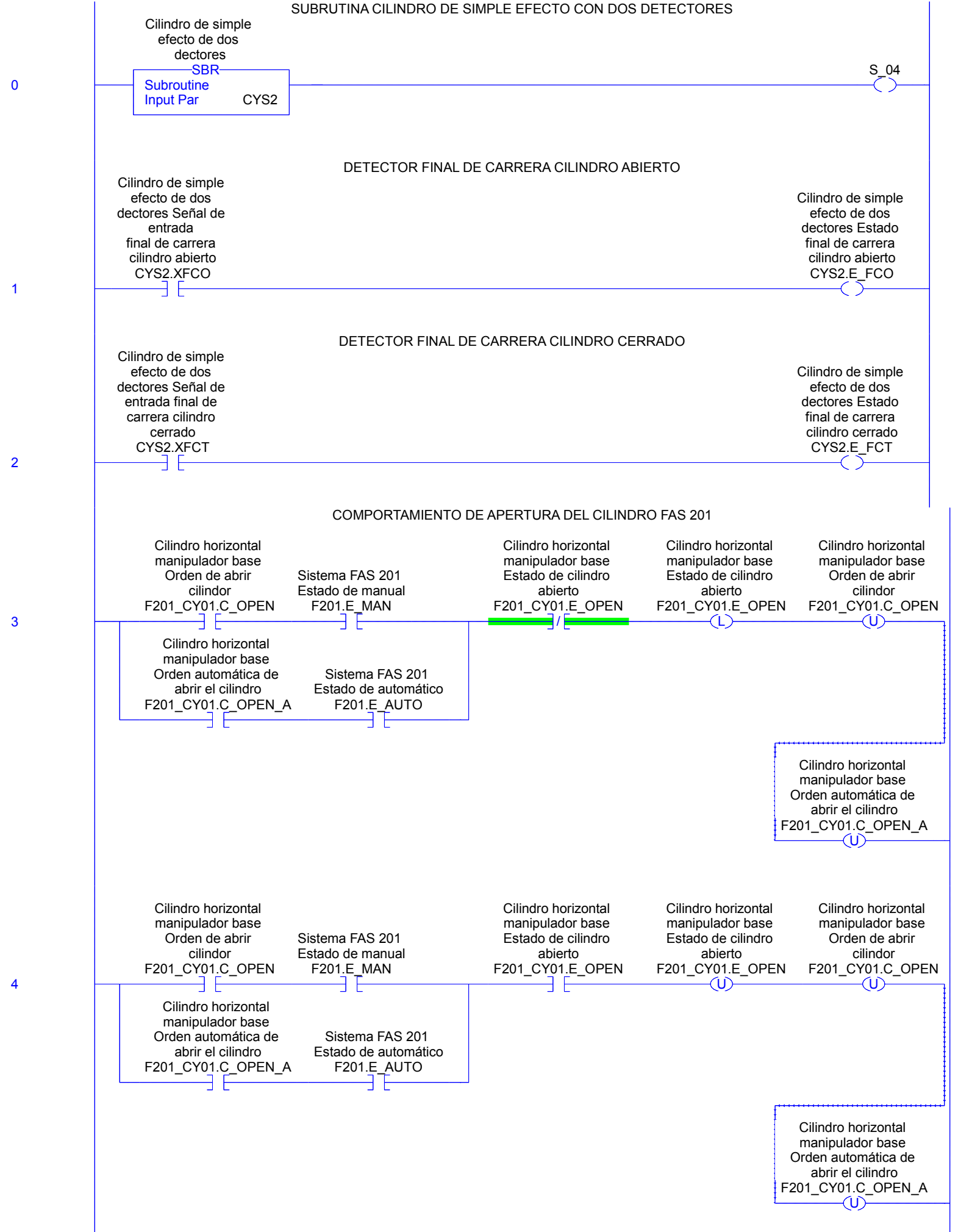


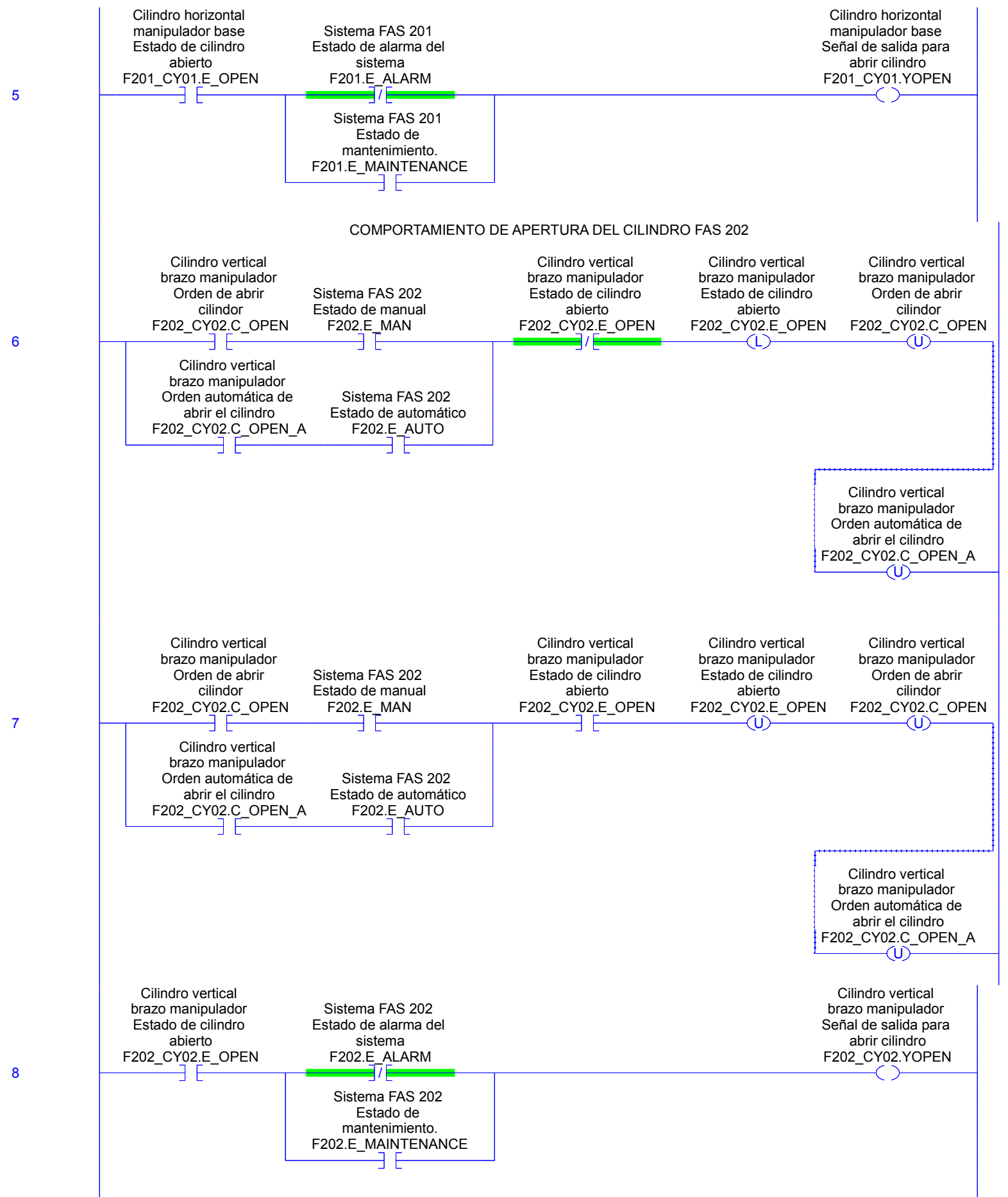


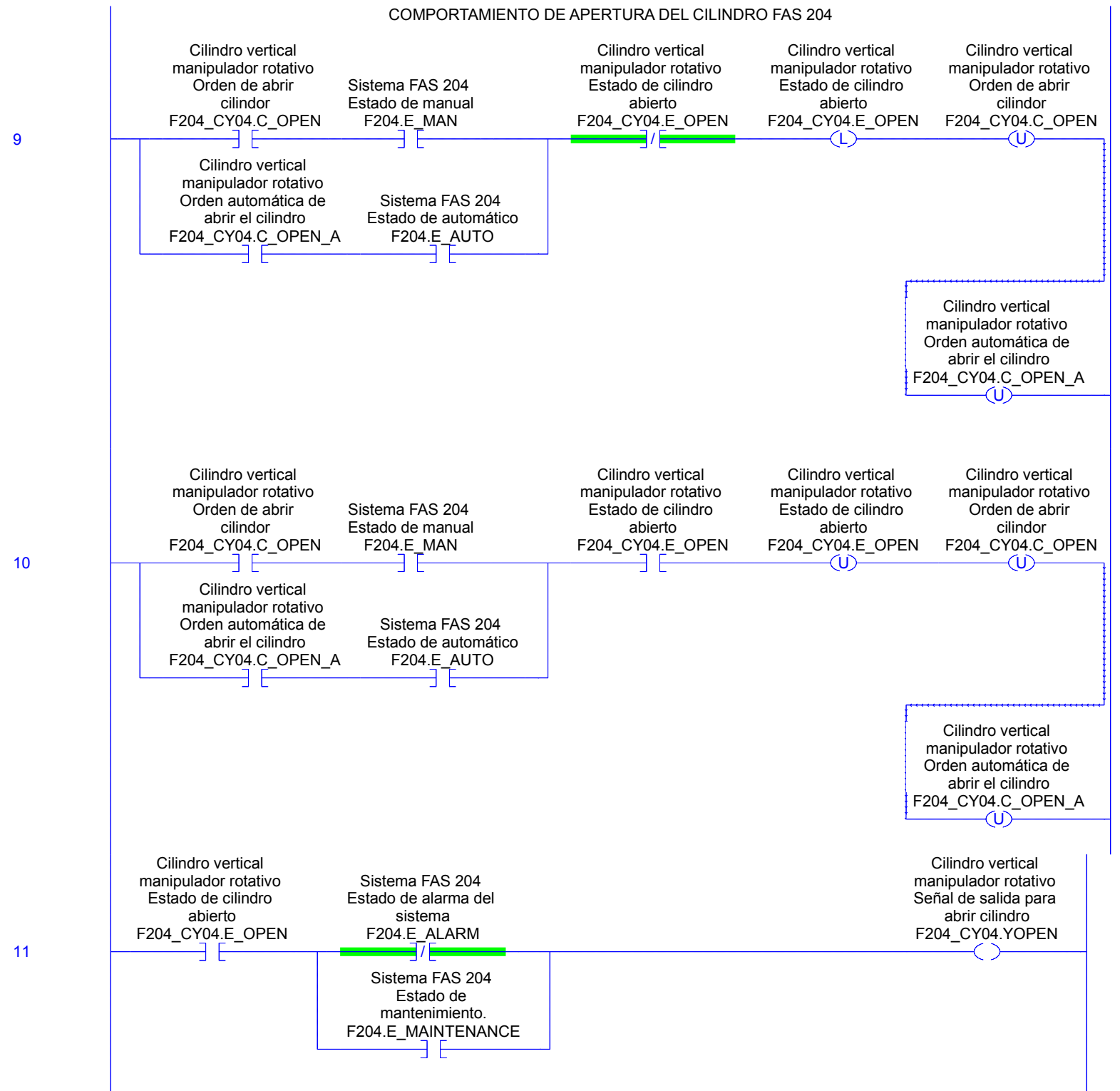


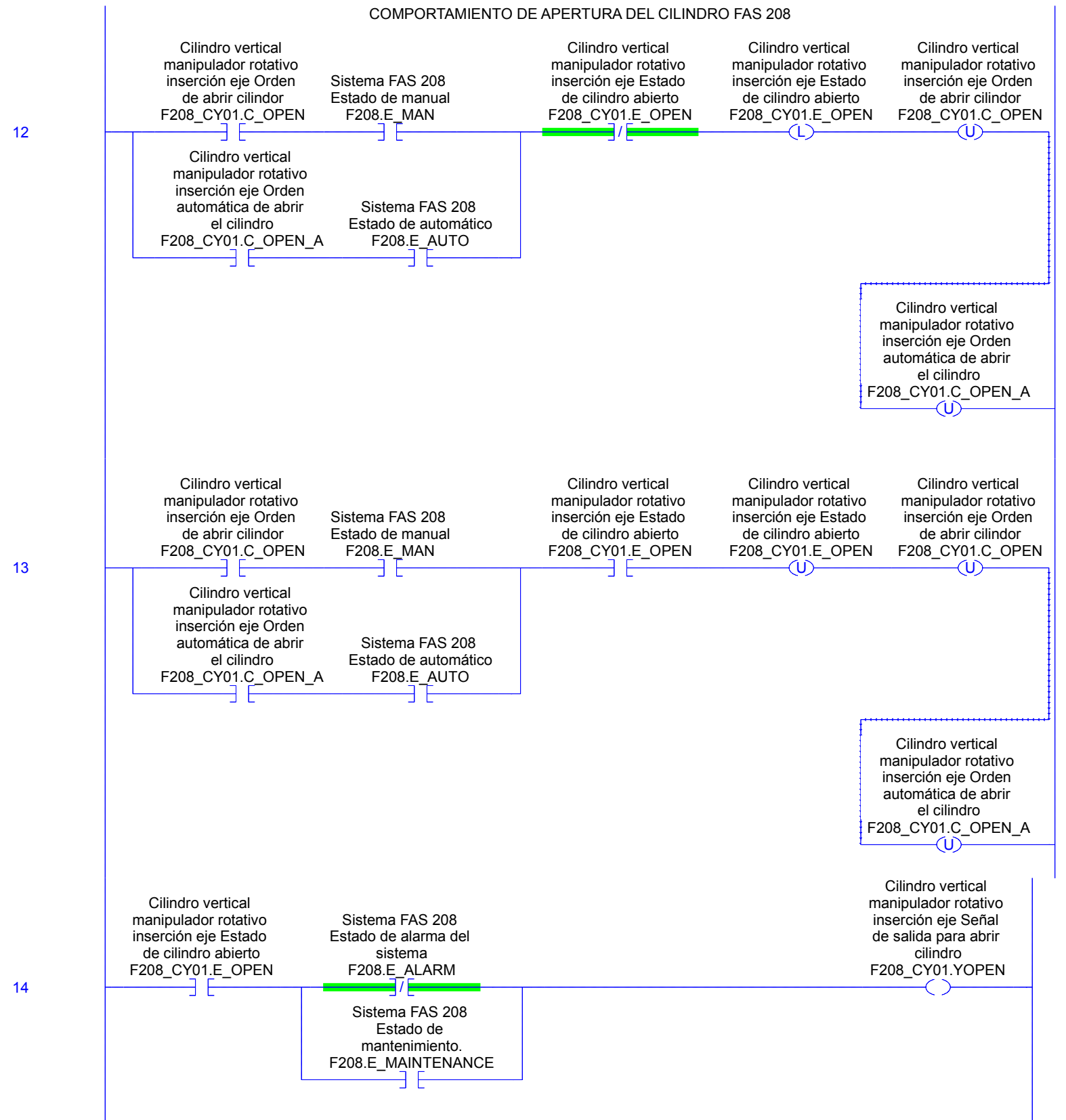


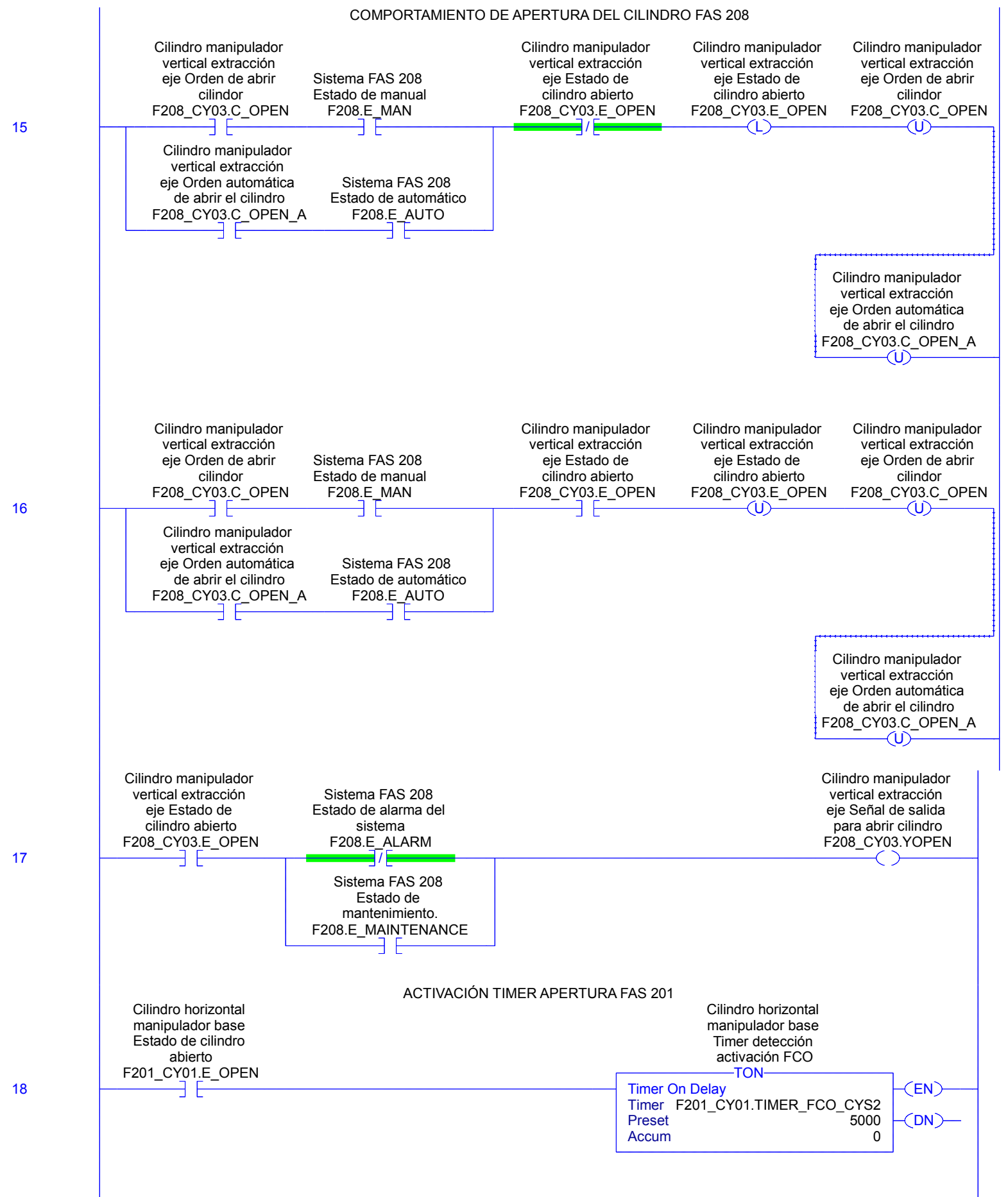


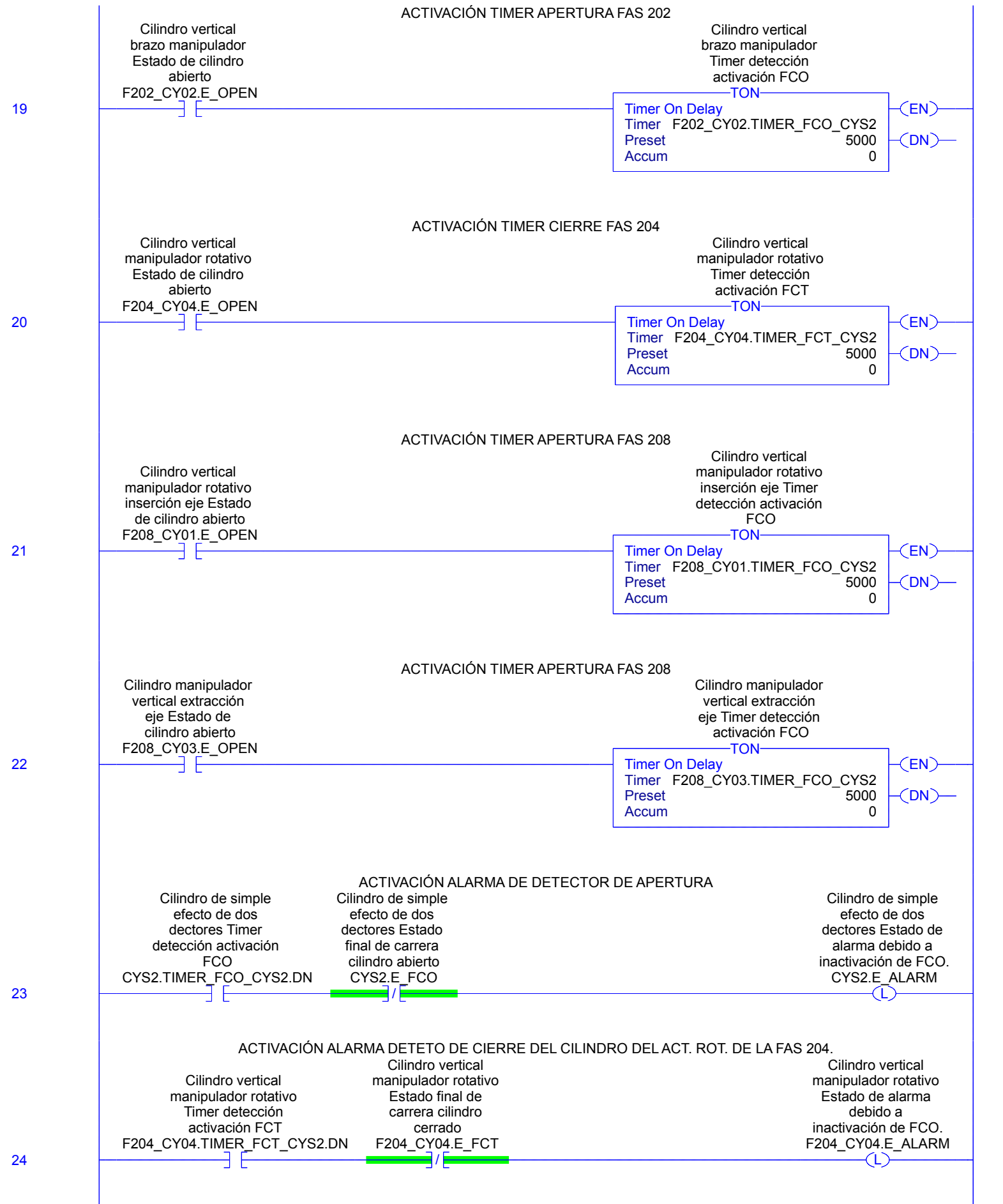


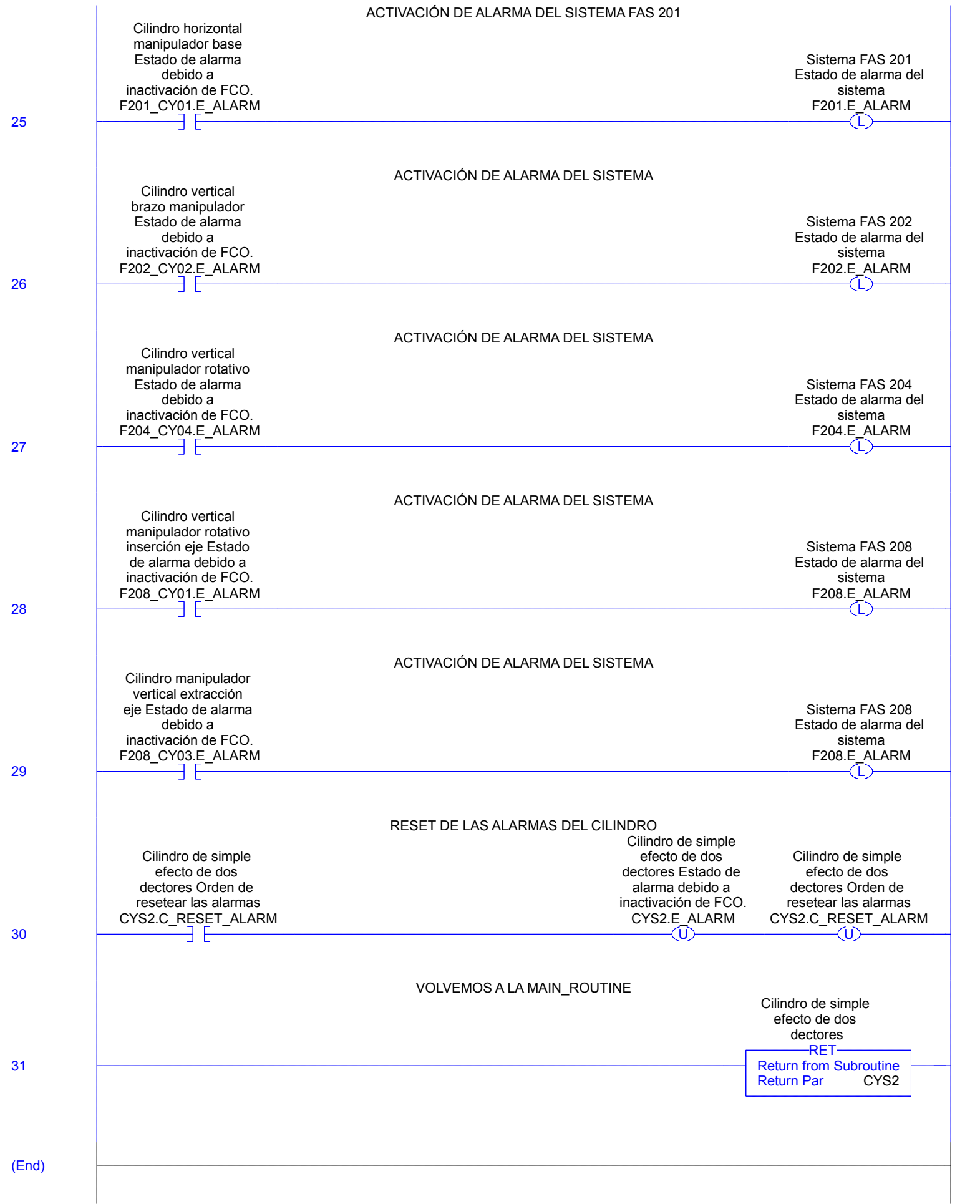


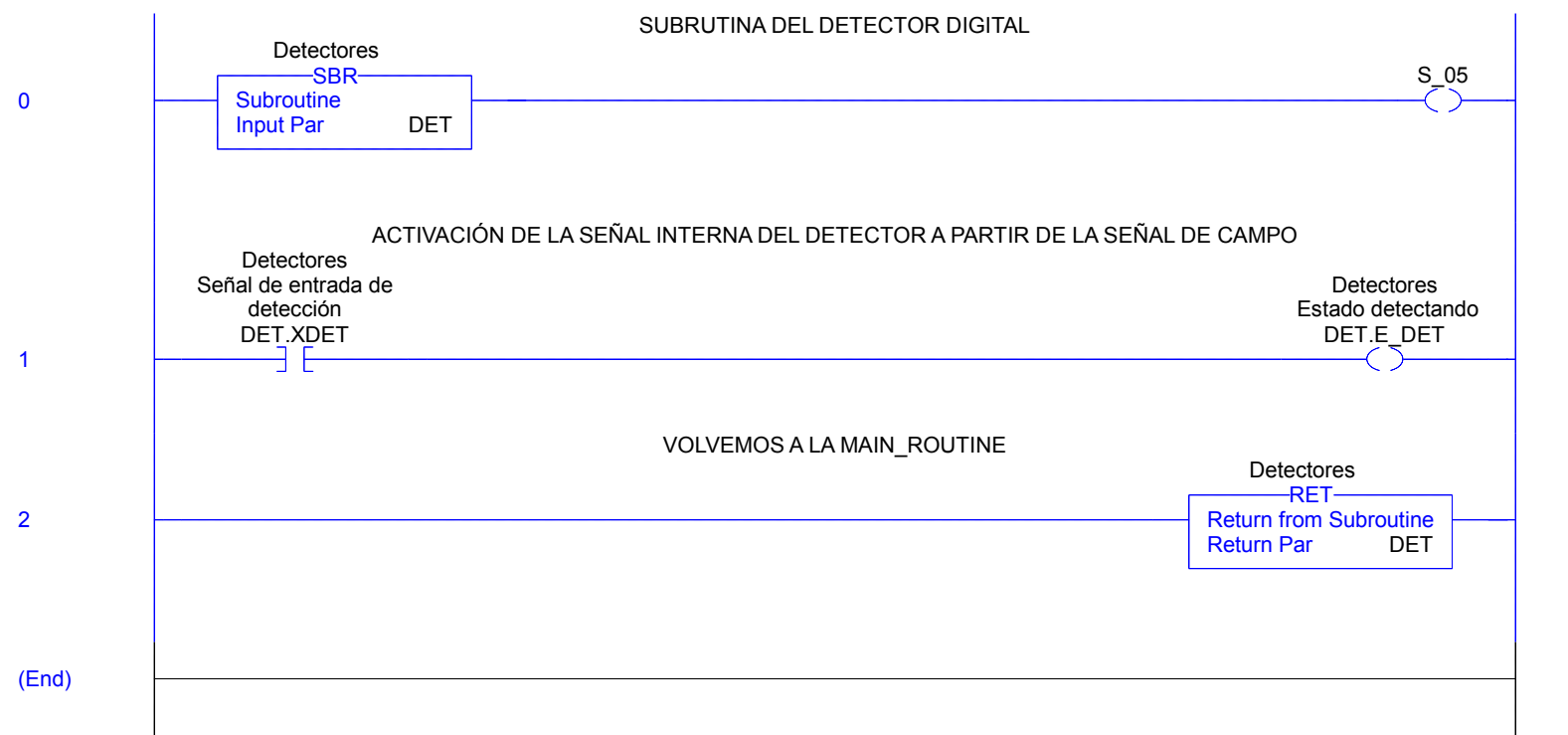


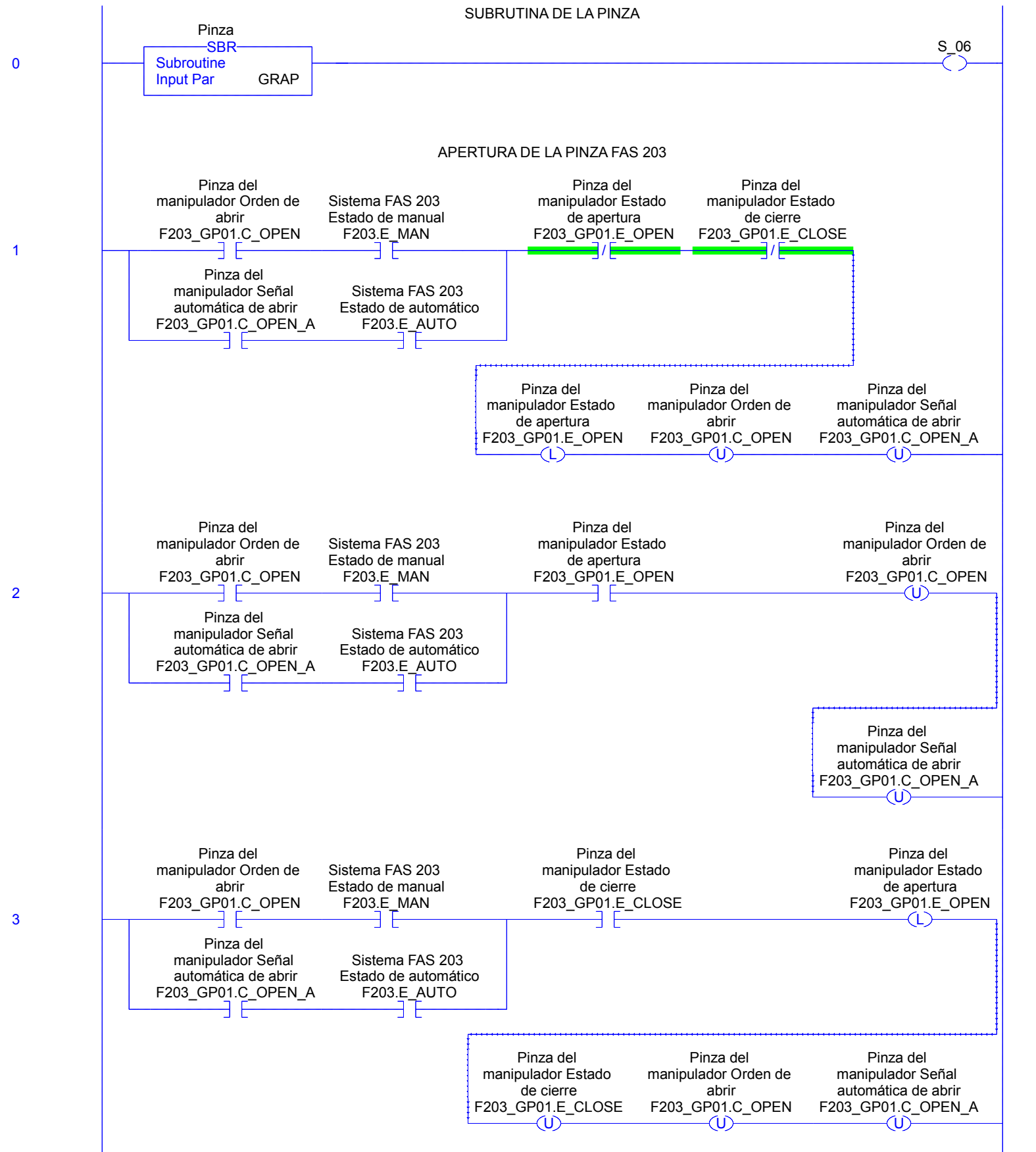












APERTURA DE LA PINZA FAS 204

4

Pinza brazo
manipulador rotativo
Orden de abrir
F204_GP01.C_OPEN

Sistema FAS 204
Estado de manual
F204.E_MAN

Pinza brazo
manipulador rotativo
Estado de apertura
F204_GP01.E_OPEN

Pinza brazo
manipulador rotativo
Estado de cierre
F204_GP01.E_CLOSE

Pinza brazo
manipulador rotativo
Señal automática de
abrir
F204_GP01.C_OPEN_A

Sistema FAS 204
Estado de automático
F204.E_AUTO

Pinza brazo
manipulador rotativo
Estado de apertura
F204_GP01.E_OPEN

Pinza brazo
manipulador rotativo
Orden de abrir
F204_GP01.C_OPEN

Pinza brazo
manipulador rotativo
Señal automática de
abrir
F204_GP01.C_OPEN_A

5

Pinza brazo
manipulador rotativo
Orden de abrir
F204_GP01.C_OPEN

Sistema FAS 204
Estado de manual
F204.E_MAN

Pinza brazo
manipulador rotativo
Estado de apertura
F204_GP01.E_OPEN

Pinza brazo
manipulador rotativo
Orden de abrir
F204_GP01.C_OPEN

Pinza brazo
manipulador rotativo
Señal automática de
abrir
F204_GP01.C_OPEN_A

Sistema FAS 204
Estado de automático
F204.E_AUTO

Pinza brazo
manipulador rotativo
Señal automática de
abrir
F204_GP01.C_OPEN_A

6

Pinza brazo
manipulador rotativo
Orden de abrir
F204_GP01.C_OPEN

Sistema FAS 204
Estado de manual
F204.E_MAN

Pinza brazo
manipulador rotativo
Estado de cierre
F204_GP01.E_CLOSE

Pinza brazo
manipulador rotativo
Estado de apertura
F204_GP01.E_OPEN

Pinza brazo
manipulador rotativo
Señal automática de
abrir
F204_GP01.C_OPEN_A

Sistema FAS 204
Estado de automático
F204.E_AUTO

Pinza brazo
manipulador rotativo
Estado de cierre
F204_GP01.E_CLOSE

Pinza brazo
manipulador rotativo
Orden de abrir
F204_GP01.C_OPEN

Pinza brazo
manipulador rotativo
Señal automática de
abrir
F204_GP01.C_OPEN_A

CIERRE DE LA PINZA FAS 203

7

Pinza del
manipulador Orden de
cerrar
F203_GP01.C_CLOSE

Sistema FAS 203
Estado de manual
F203.E_MAN

Pinza del
manipulador Estado
de apertura
F203_GP01.E_OPEN

Pinza del
manipulador Estado
de cierre
F203_GP01.E_CLOSE

Pinza del
manipulador Señal
automática de cerrar
F203_GP01.C_CLOSE_A

Sistema FAS 203
Estado de automático
F203.E_AUTO

Pinza del
manipulador Estado
de cierre
F203_GP01.E_CLOSE

Pinza del
manipulador Orden de
cerrar
F203_GP01.C_CLOSE

Pinza del
manipulador Señal
automática de cerrar
F203_GP01.C_CLOSE_A

8

Pinza del
manipulador Orden de
cerrar
F203_GP01.C_CLOSE

Sistema FAS 203
Estado de manual
F203.E_MAN

Pinza del
manipulador Estado
de cierre
F203_GP01.E_CLOSE

Pinza del
manipulador Orden de
cerrar
F203_GP01.C_CLOSE

Pinza del
manipulador Señal
automática de cerrar
F203_GP01.C_CLOSE_A

Sistema FAS 203
Estado de automático
F203.E_AUTO

Pinza del
manipulador Señal
automática de cerrar
F203_GP01.C_CLOSE_A

9

Pinza del
manipulador Orden de
cerrar
F203_GP01.C_CLOSE

Sistema FAS 203
Estado de manual
F203.E_MAN

Pinza del
manipulador Estado
de apertura
F203_GP01.E_OPEN

Pinza del
manipulador Estado
de cierre
F203_GP01.E_CLOSE

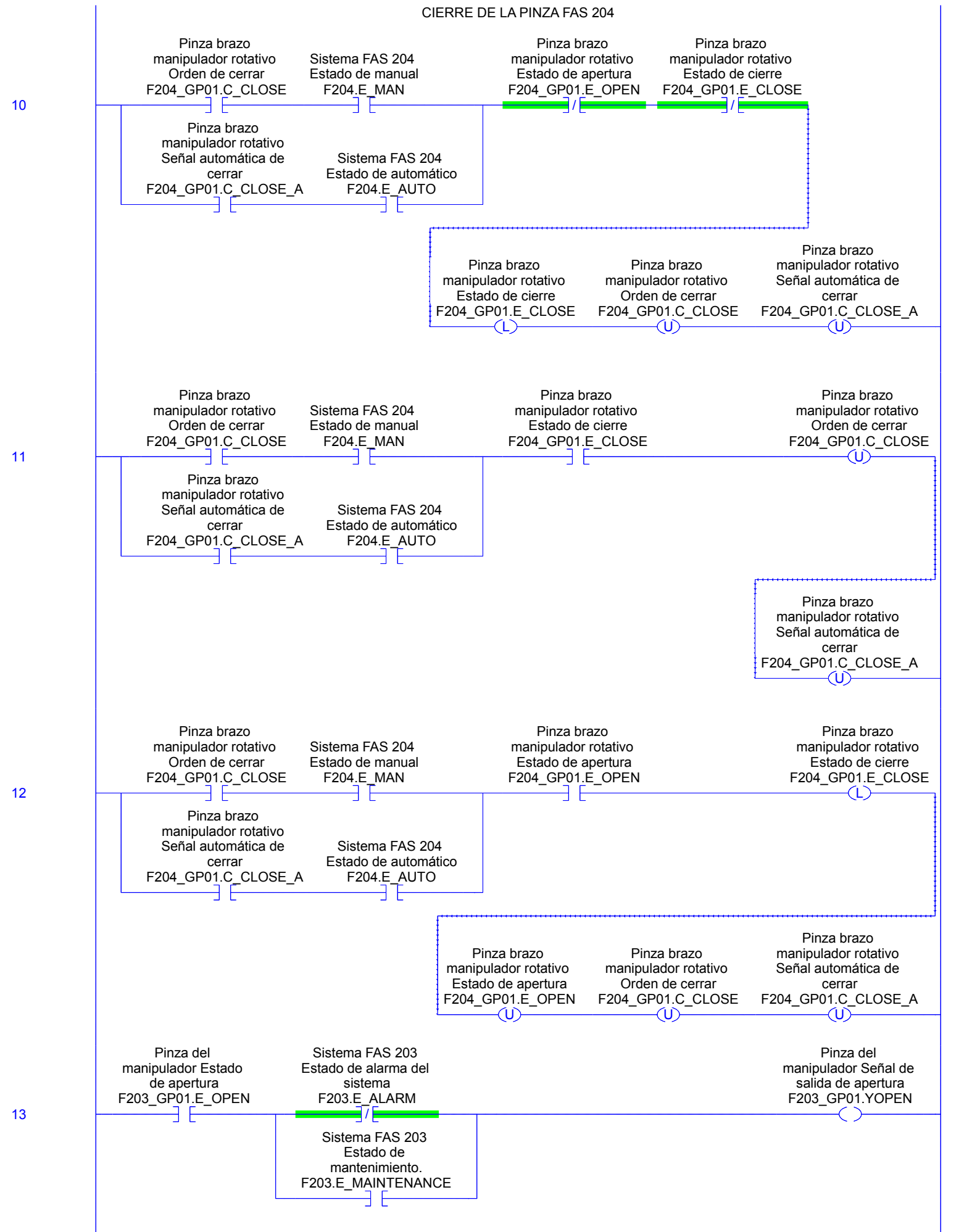
Pinza del
manipulador Señal
automática de cerrar
F203_GP01.C_CLOSE_A

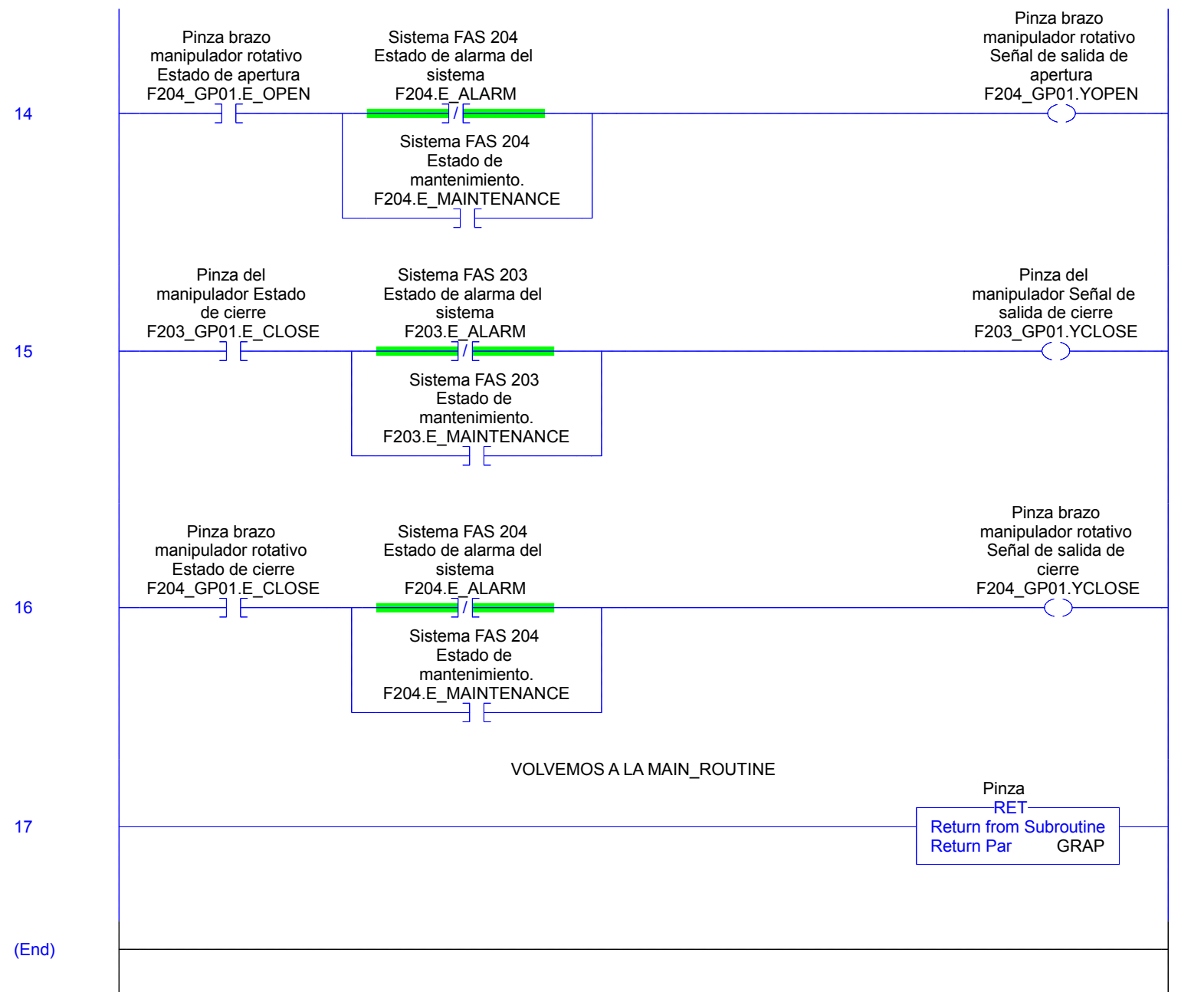
Sistema FAS 203
Estado de automático
F203.E_AUTO

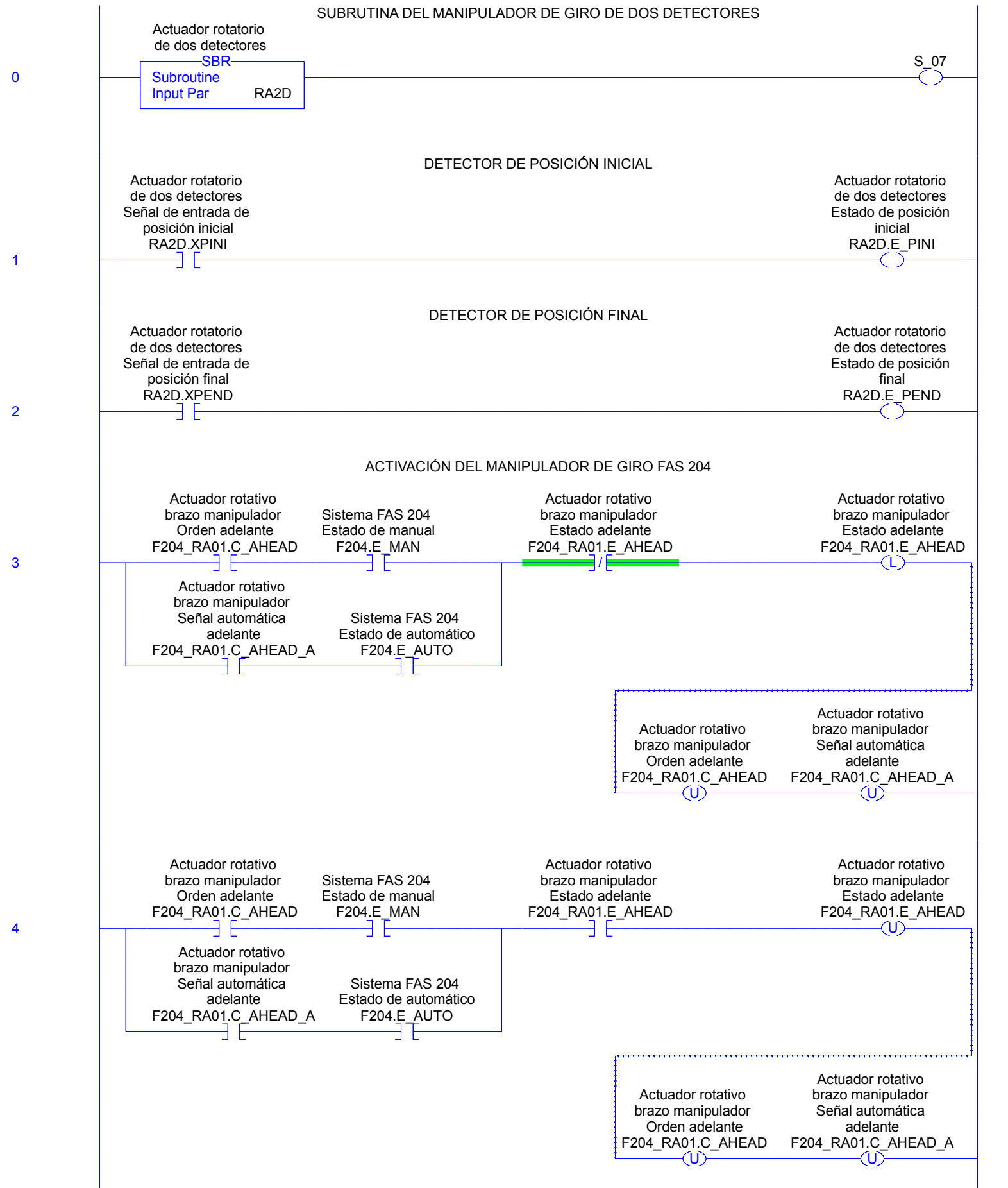
Pinza del
manipulador Estado
de apertura
F203_GP01.E_OPEN

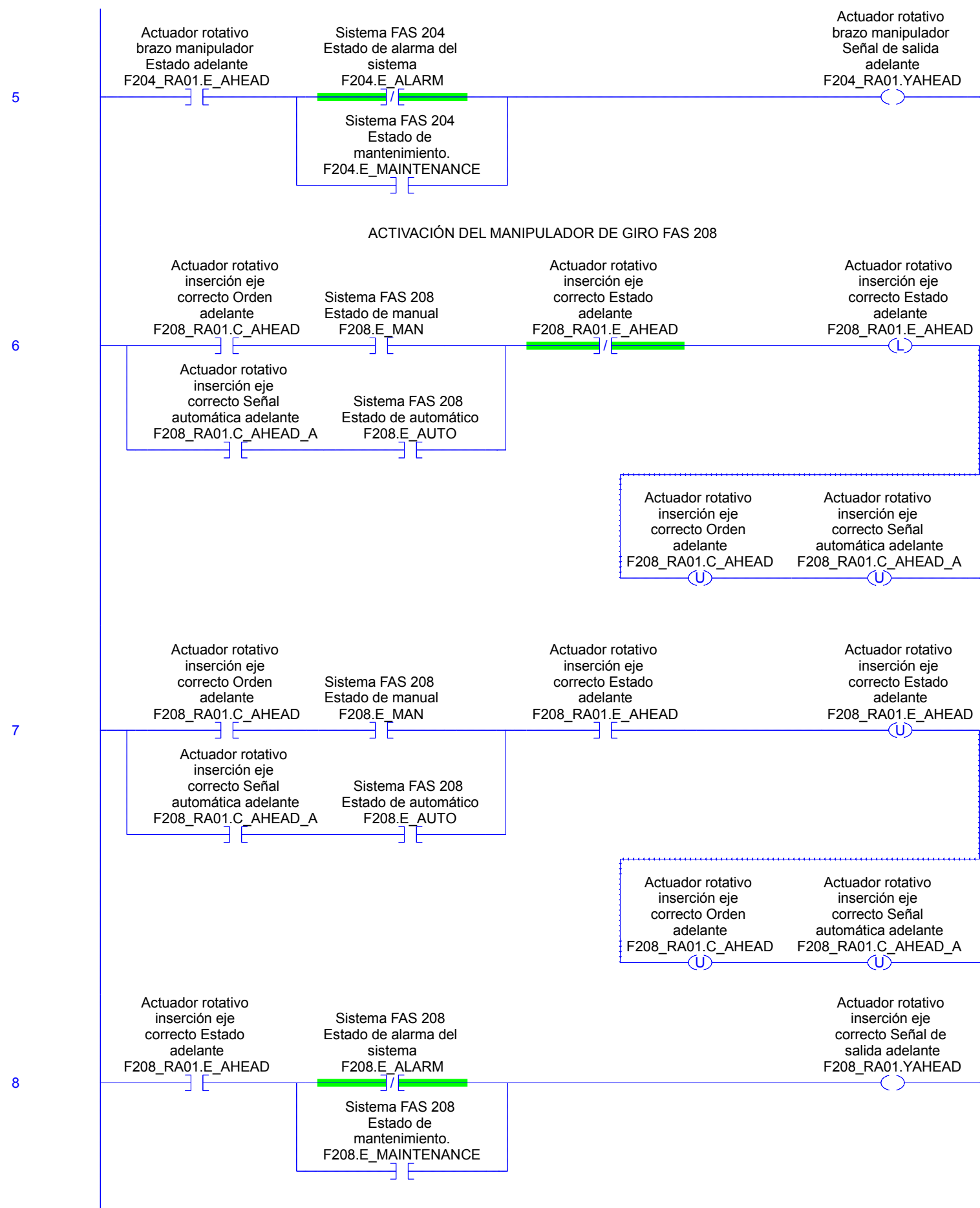
Pinza del
manipulador Orden de
cerrar
F203_GP01.C_CLOSE

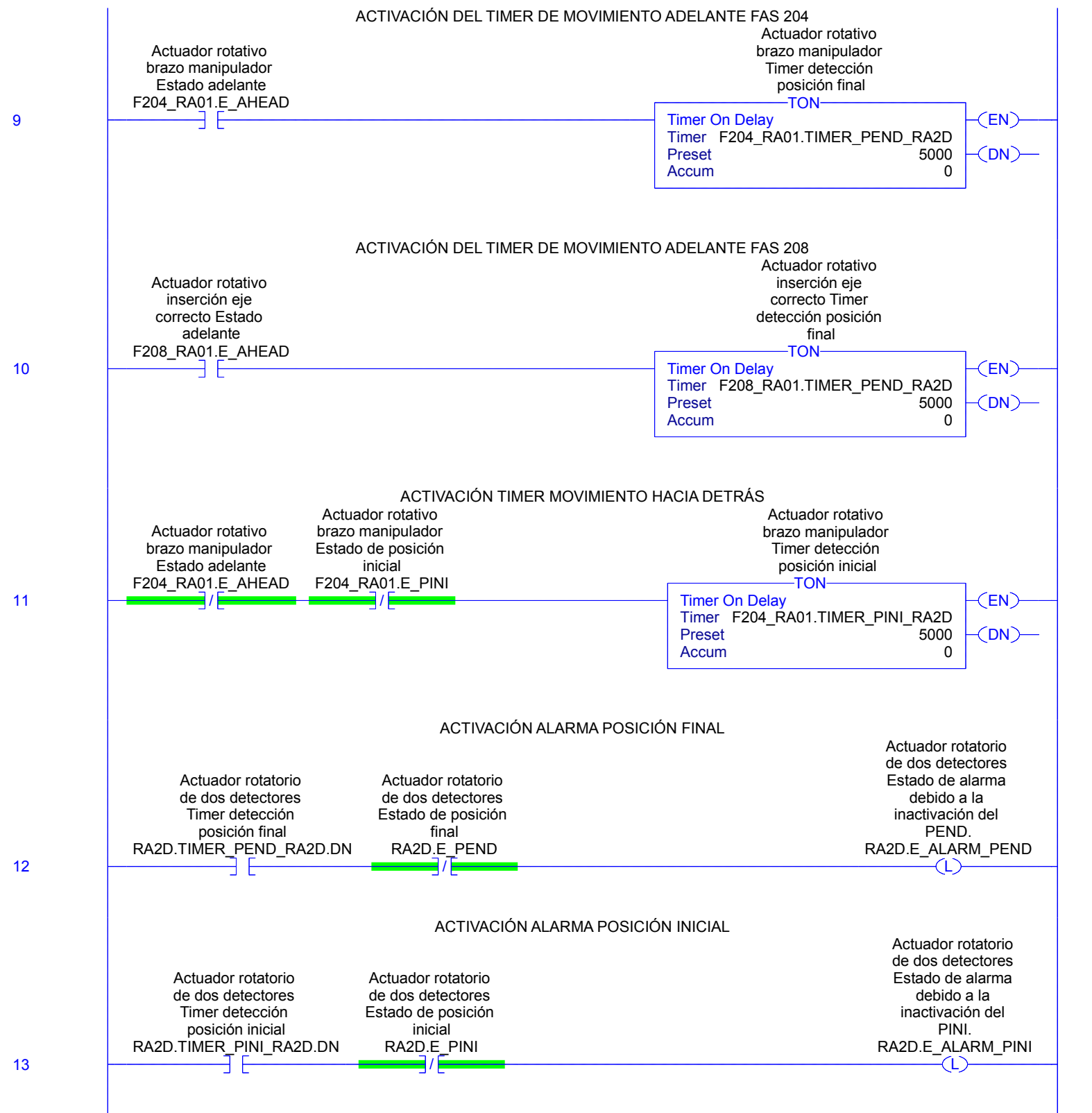
Pinza del
manipulador Señal
automática de cerrar
F203_GP01.C_CLOSE_A

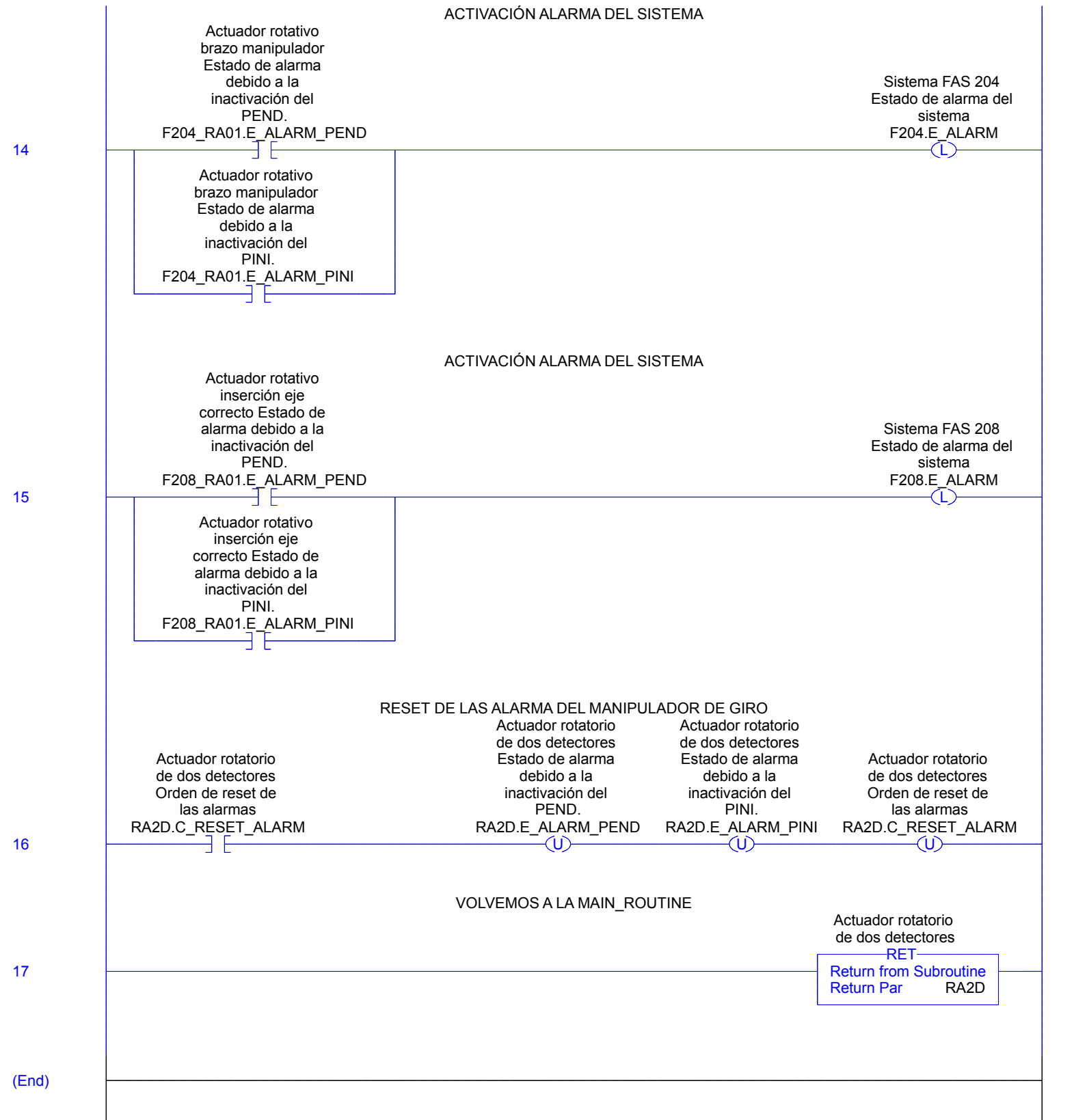


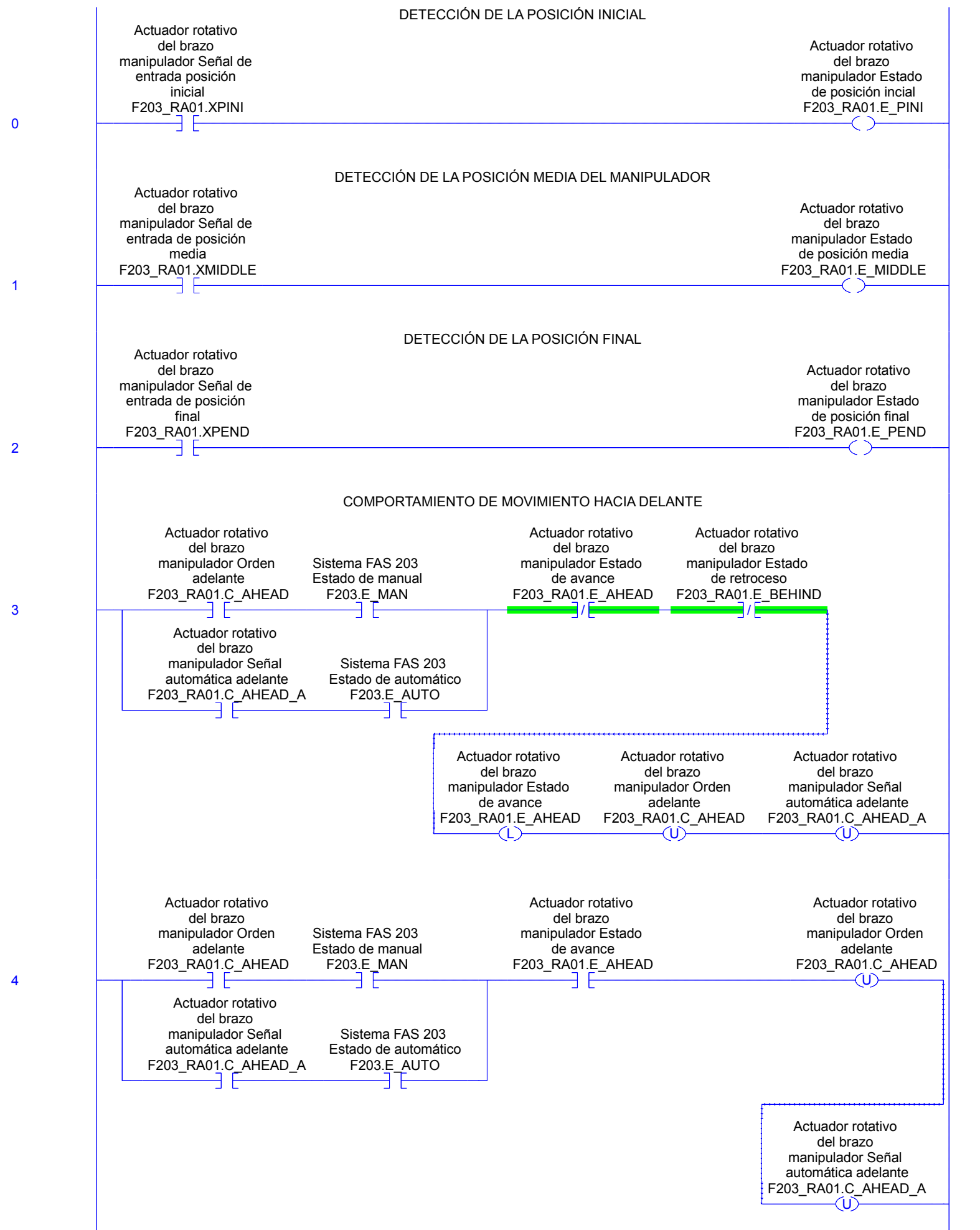


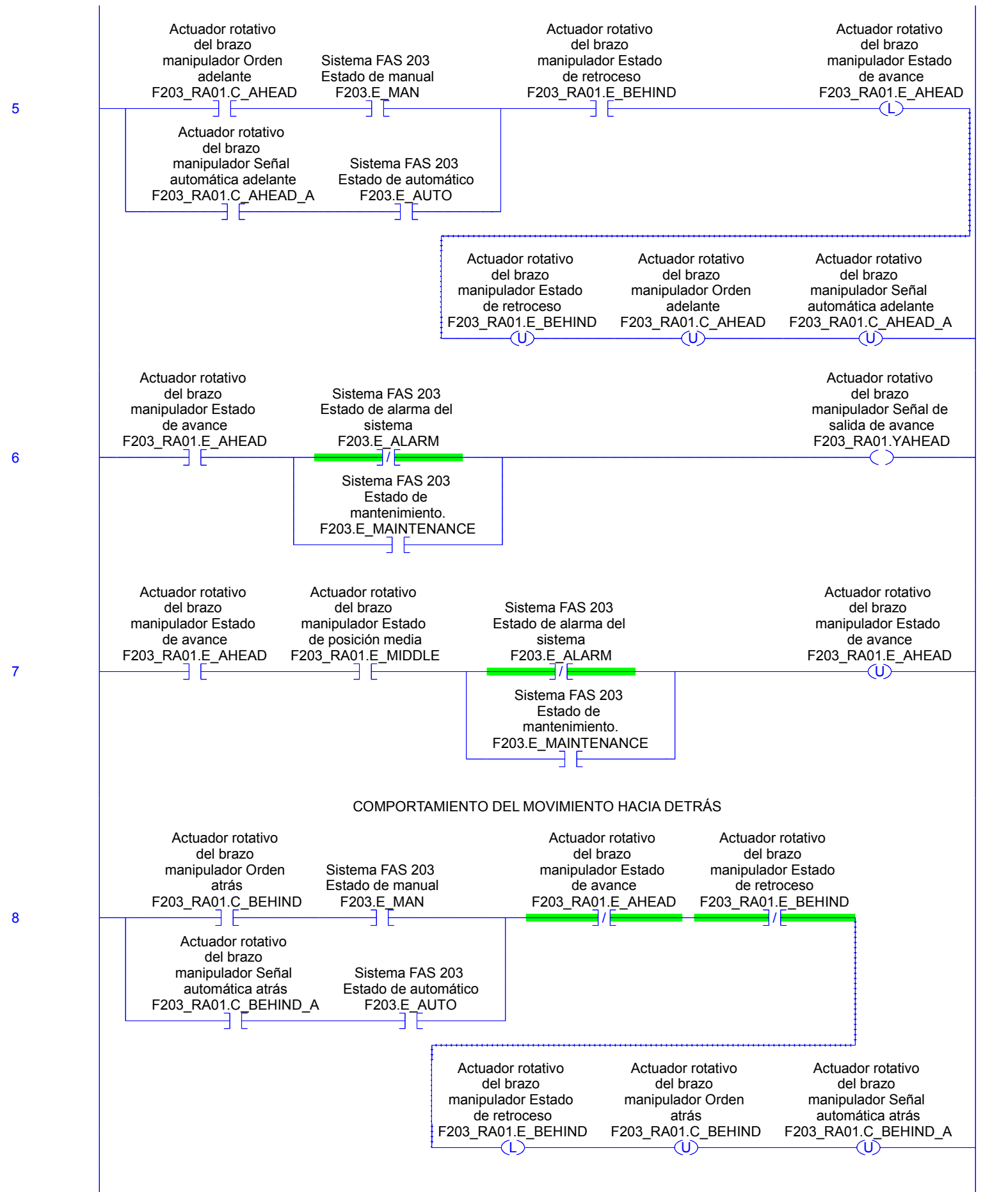


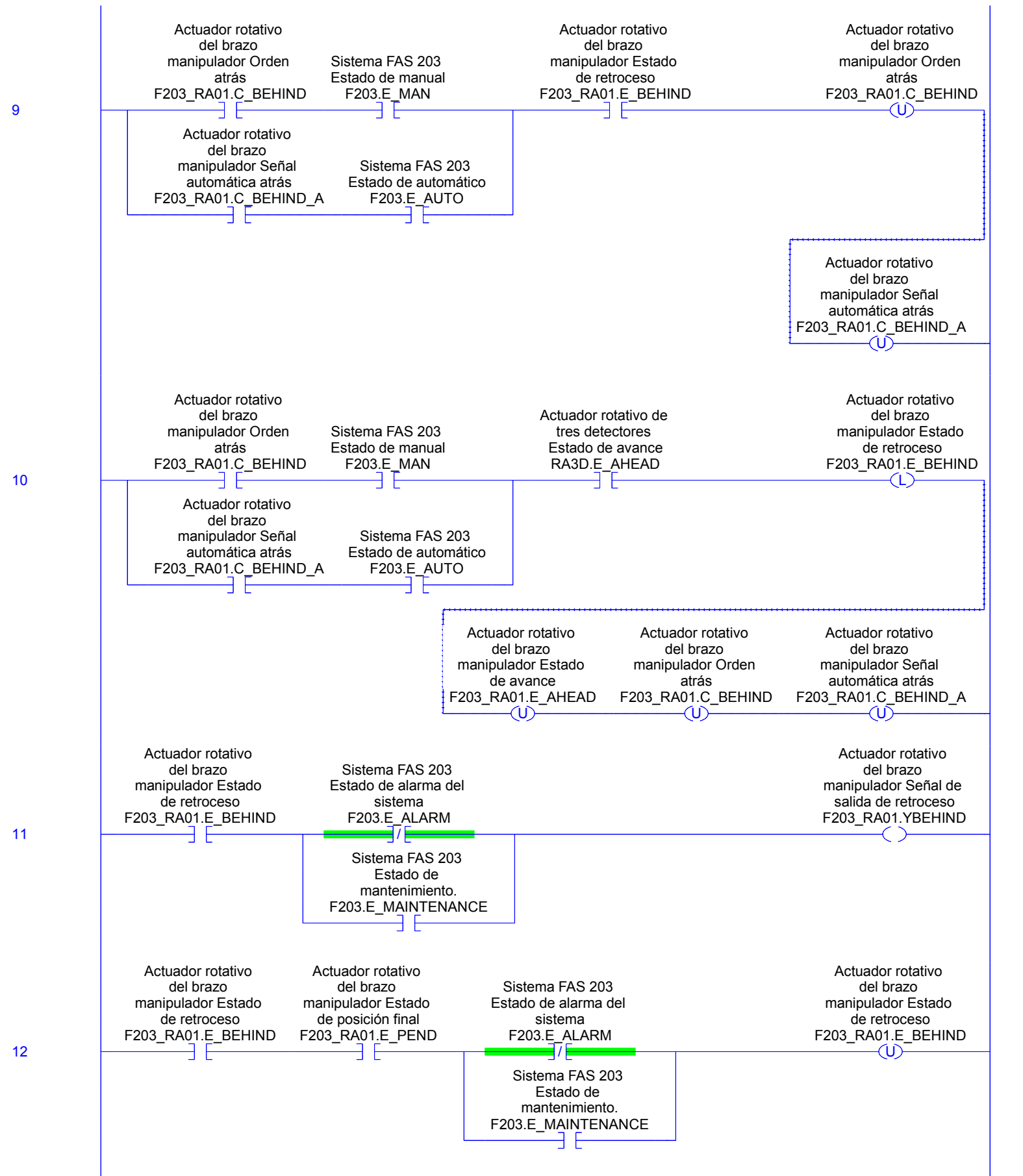


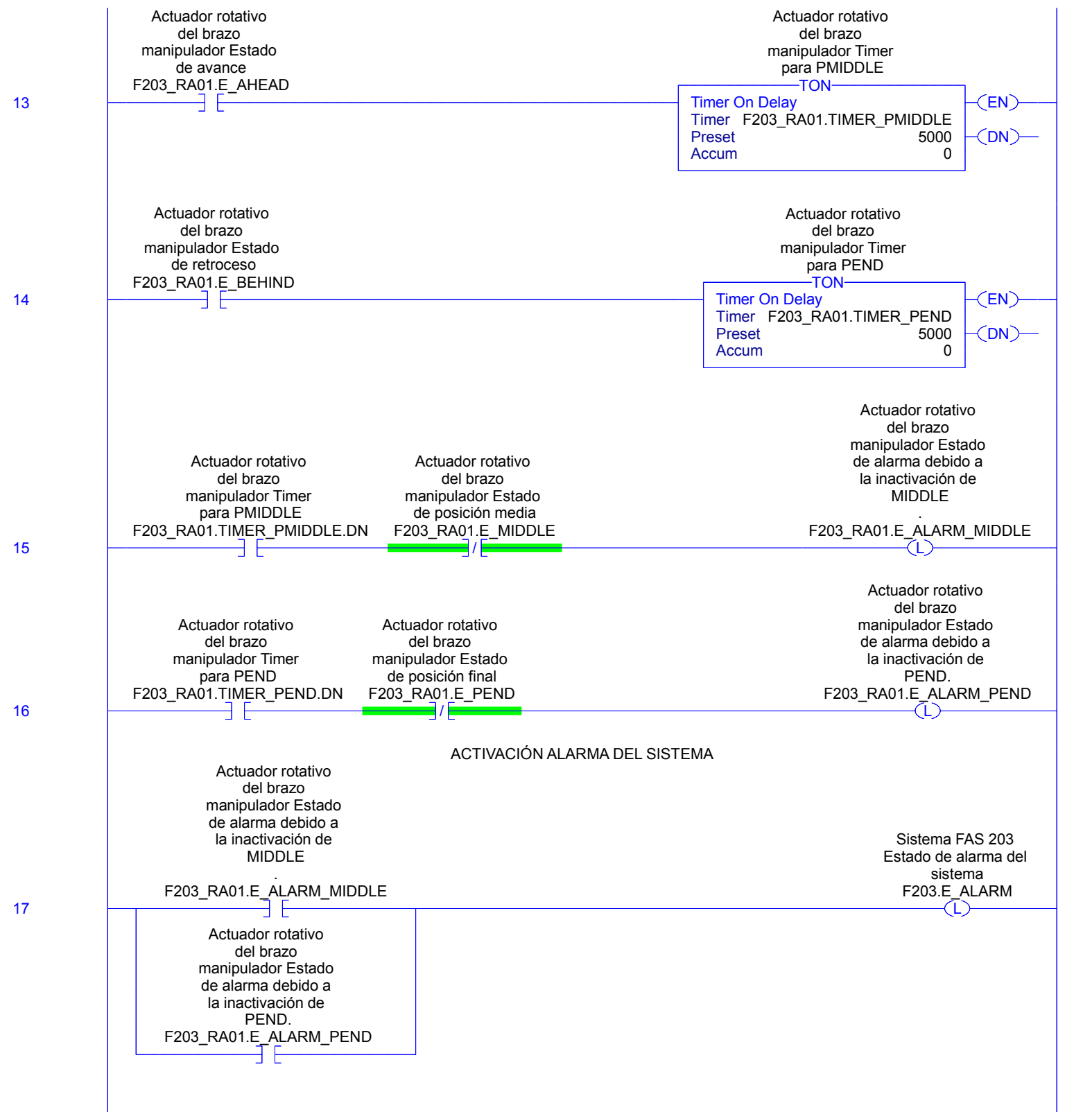


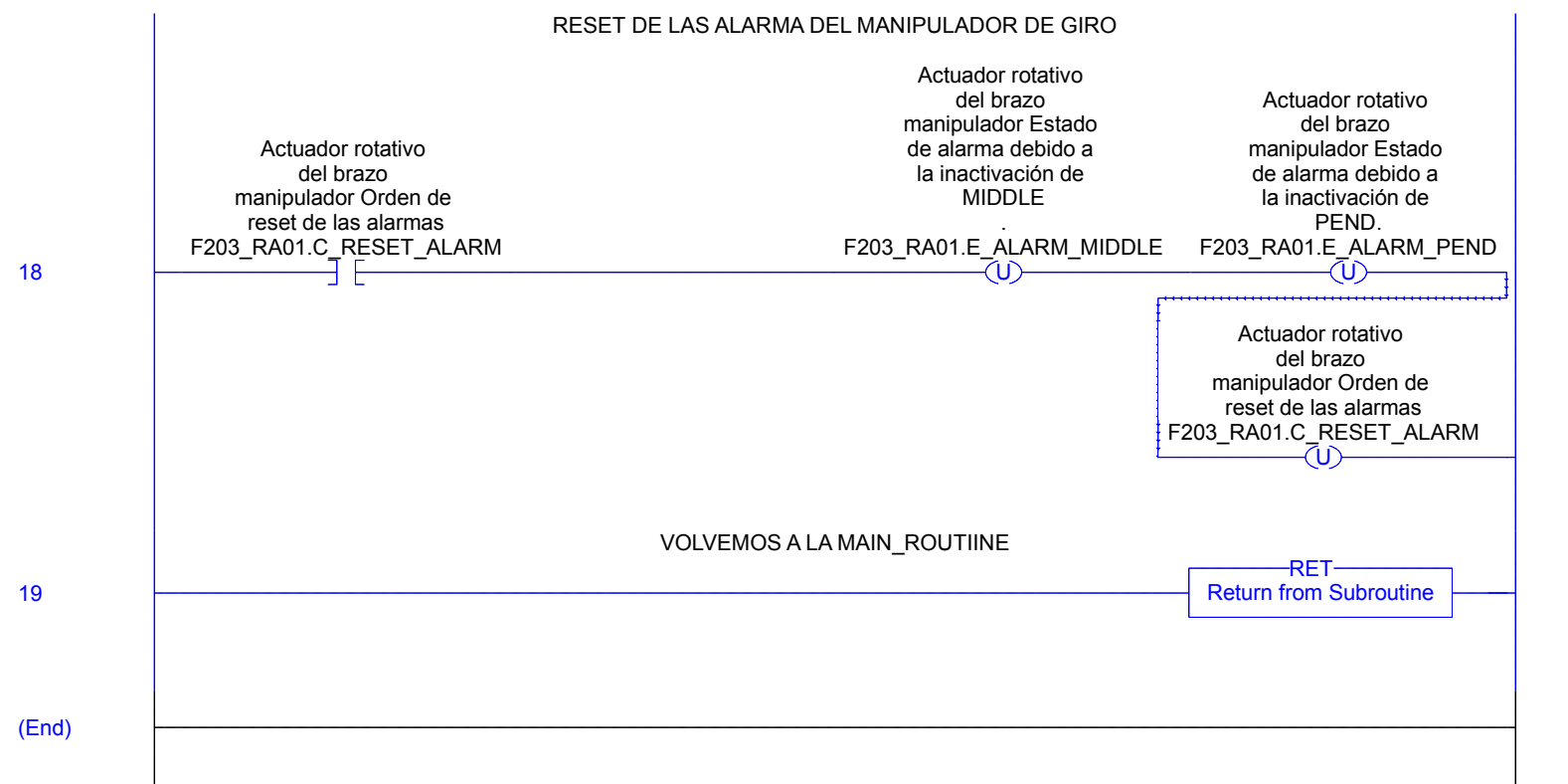


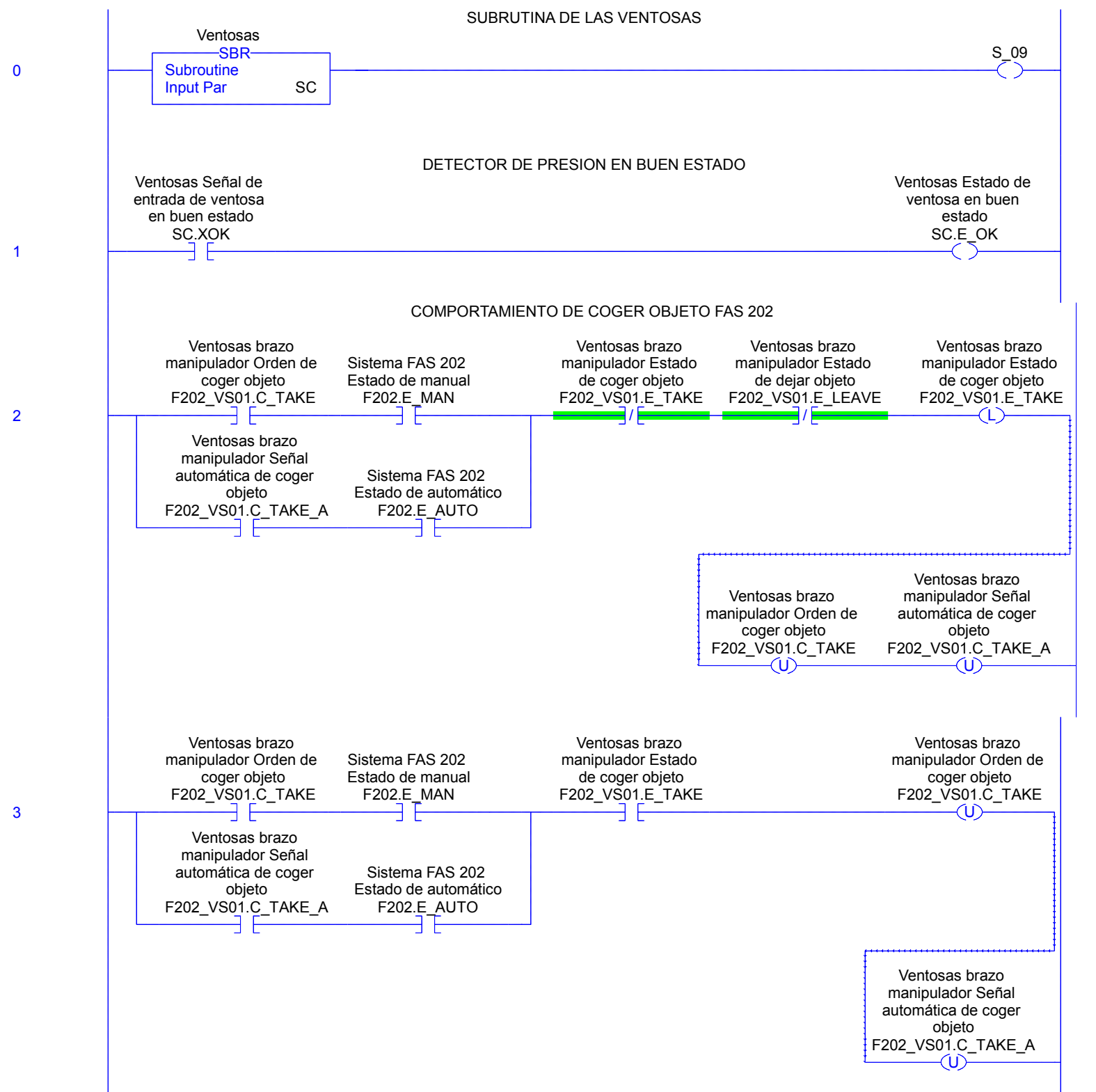


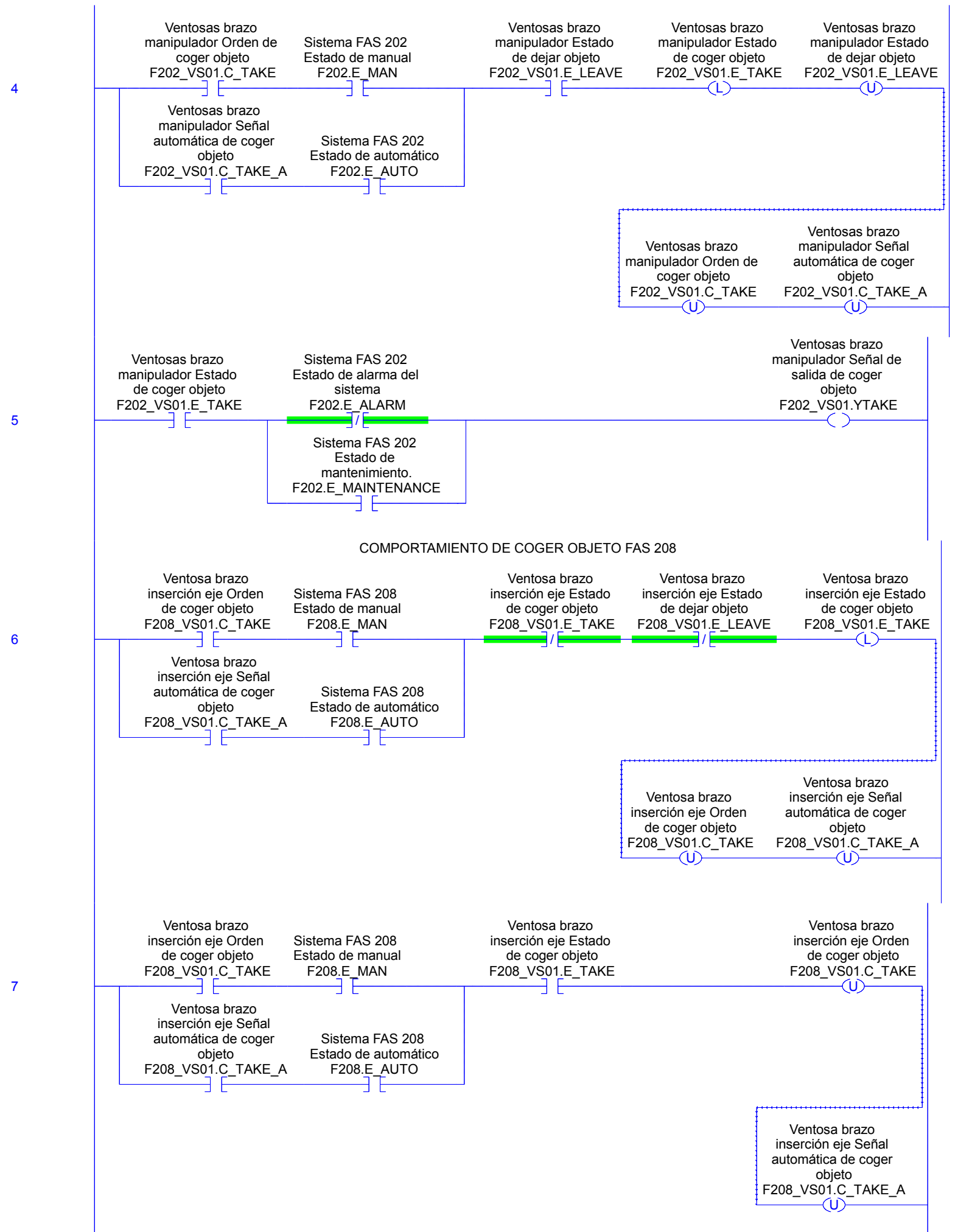


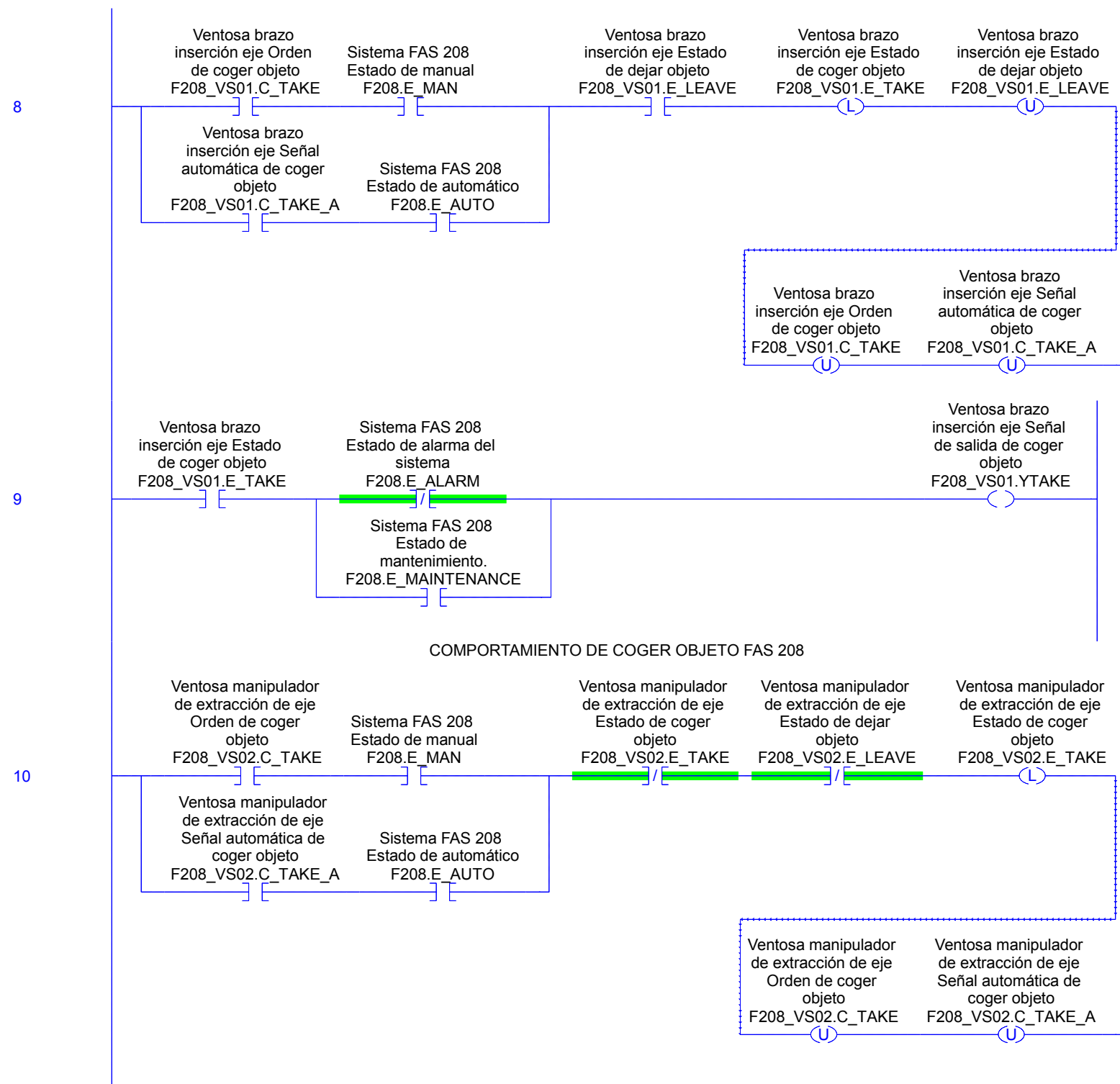


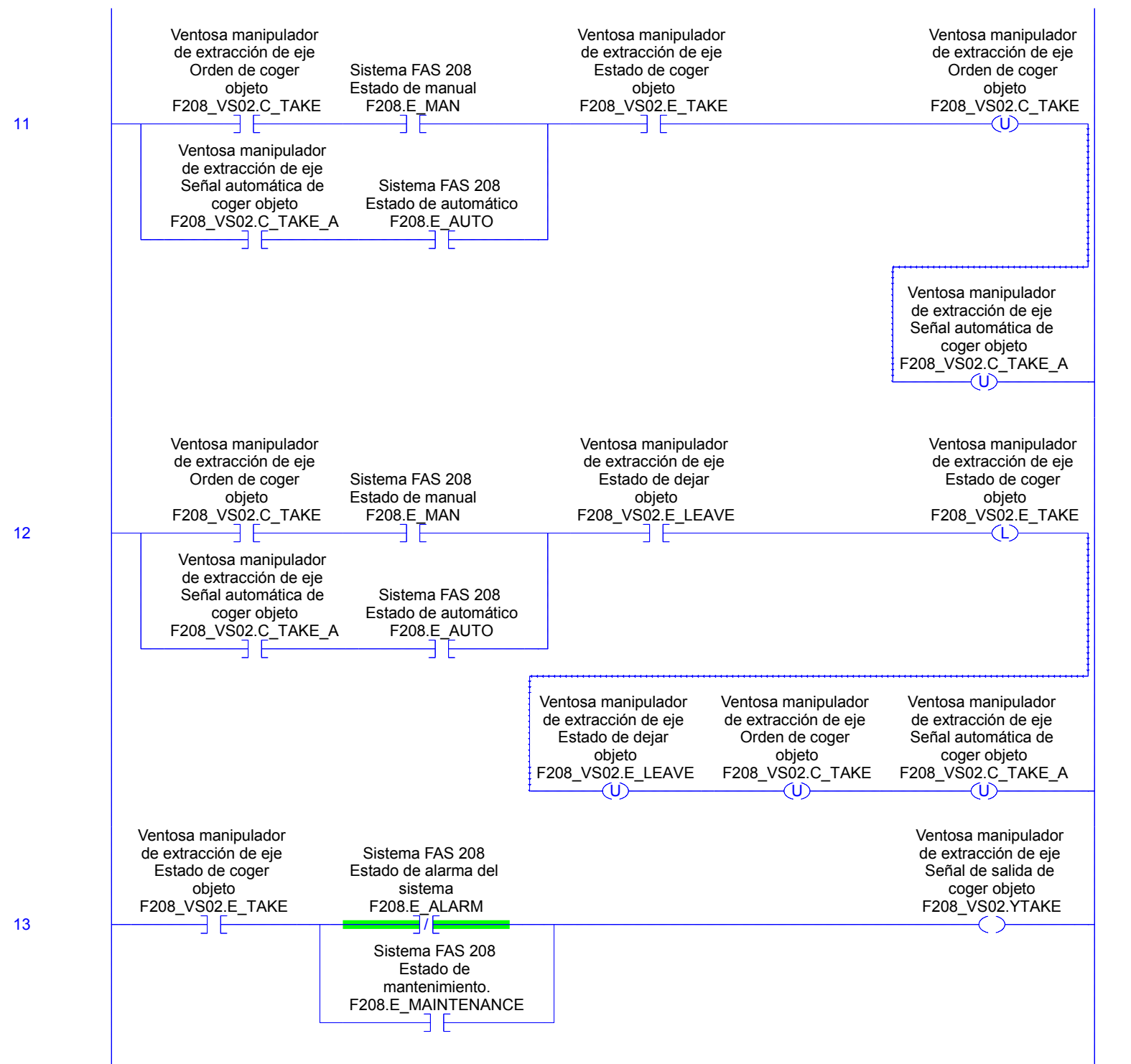












COMPORTAMIENTO DE DEJAR OBJETO FAS 202

14

Ventosas brazo
manipulador Orden de
dejar objeto
F202_VS01.C_LEAVE

Sistema FAS 202
Estado de manual
F202.E_MAN

Ventosas brazo
manipulador Estado
de coger objeto
F202_VS01.E_TAKE

Ventosas brazo
manipulador Estado
de dejar objeto
F202_VS01.E_LEAVE

Ventosas brazo
manipulador Estado
de coger objeto
F202_VS01.E_TAKE

Ventosas brazo
manipulador Señal
automática de dejar
objeto
F202_VS01.C_LEAVE_A

Sistema FAS 202
Estado de automático
F202.E_AUTO

Ventosas brazo
manipulador Orden de
dejar objeto
F202_VS01.C_LEAVE

Ventosas brazo
manipulador Señal
automática de dejar
objeto
F202_VS01.C_LEAVE_A

15

Ventosas brazo
manipulador Orden de
dejar objeto
F202_VS01.C_LEAVE

Sistema FAS 202
Estado de manual
F202.E_MAN

Ventosas brazo
manipulador Estado
de dejar objeto
F202_VS01.E_LEAVE

Ventosas brazo
manipulador Orden de
dejar objeto
F202_VS01.C_LEAVE

Ventosas brazo
manipulador Señal
automática de dejar
objeto
F202_VS01.C_LEAVE_A

Sistema FAS 202
Estado de automático
F202.E_AUTO

Ventosas brazo
manipulador Señal
automática de dejar
objeto
F202_VS01.C_LEAVE_A

16

Ventosas brazo
manipulador Orden de
dejar objeto
F202_VS01.C_LEAVE

Sistema FAS 202
Estado de manual
F202.E_MAN

Ventosas brazo
manipulador Estado
de coger objeto
F202_VS01.E_TAKE

Ventosas brazo
manipulador Estado
de dejar objeto
F202_VS01.E_LEAVE

Ventosas brazo
manipulador Estado
de coger objeto
F202_VS01.E_TAKE

Ventosas brazo
manipulador Señal
automática de dejar
objeto
F202_VS01.C_LEAVE_A

Sistema FAS 202
Estado de automático
F202.E_AUTO

Ventosas brazo
manipulador Orden de
dejar objeto
F202_VS01.C_LEAVE

Ventosas brazo
manipulador Señal
automática de dejar
objeto
F202_VS01.C_LEAVE_A

17

Ventosas brazo
manipulador Estado
de dejar objeto
F202_VS01.E_LEAVE

Sistema FAS 202
Estado de alarma del
sistema
F202.E_ALARM

Sistema FAS 202
Estado de
mantenimiento.
F202.E_MAINTENANCE

Ventosas brazo
manipulador Señal de
salida de dejar
objeto
F202_VS01.YLEAVE

COMPORTAMIENTO DE DEJAR OBJETO FAS 208

18

Ventosa brazo
inserción eje Orden
de dejar objeto
F208_VS01.C_LEAVE

Sistema FAS 208
Estado de manual
F208.E_MAN

Ventosa brazo
inserción eje Estado
de coger objeto
F208_VS01.E_TAKE

Ventosa brazo
inserción eje Estado
de dejar objeto
F208_VS01.E_LEAVE

Ventosa brazo
inserción eje Señal
automática de dejar
objeto
F208_VS01.C_LEAVE_A

Sistema FAS 208
Estado de automático
F208.E_AUTO

Ventosa brazo
inserción eje Estado
de dejar objeto
F208_VS01.E_LEAVE

Ventosa brazo
inserción eje Orden
de dejar objeto
F208_VS01.C_LEAVE

Ventosa brazo
inserción eje Señal
automática de dejar
objeto
F208_VS01.C_LEAVE_A

19

Ventosa brazo
inserción eje Orden
de dejar objeto
F208_VS01.C_LEAVE

Sistema FAS 208
Estado de manual
F208.E_MAN

Ventosa brazo
inserción eje Estado
de dejar objeto
F208_VS01.E_LEAVE

Ventosa brazo
inserción eje Orden
de dejar objeto
F208_VS01.C_LEAVE

Ventosa brazo
inserción eje Señal
automática de dejar
objeto
F208_VS01.C_LEAVE_A

Sistema FAS 208
Estado de automático
F208.E_AUTO

Ventosa brazo
inserción eje Señal
automática de dejar
objeto
F208_VS01.C_LEAVE_A

20

Ventosa brazo
inserción eje Orden
de dejar objeto
F208_VS01.C_LEAVE

Sistema FAS 208
Estado de manual
F208.E_MAN

Ventosa brazo
inserción eje Estado
de coger objeto
F208_VS01.E_TAKE

Ventosa brazo
inserción eje Estado
de dejar objeto
F208_VS01.E_LEAVE

Ventosa brazo
inserción eje Estado
de coger objeto
F208_VS01.E_TAKE

Ventosa brazo
inserción eje Señal
automática de dejar
objeto
F208_VS01.C_LEAVE_A

Sistema FAS 208
Estado de automático
F208.E_AUTO

Ventosa brazo
inserción eje Orden
de dejar objeto
F208_VS01.C_LEAVE

Ventosa brazo
inserción eje Señal
automática de dejar
objeto
F208_VS01.C_LEAVE_A

21

Ventosa brazo
inserción eje Estado
de dejar objeto
F208_VS01.E_LEAVE

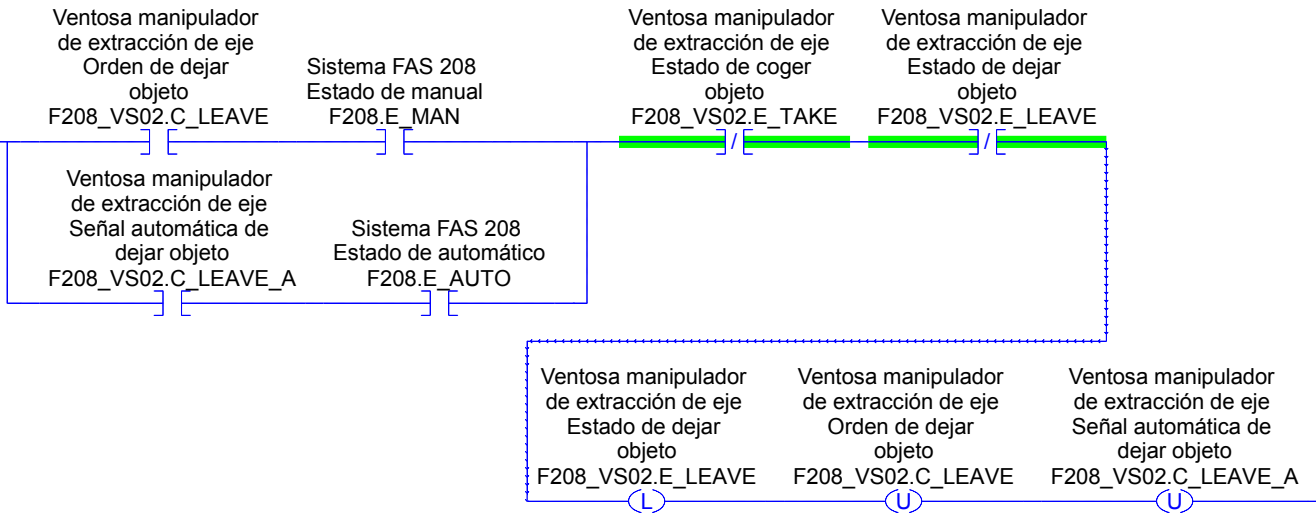
Sistema FAS 208
Estado de alarma del
sistema
F208.E_ALARM

Sistema FAS 208
Estado de
mantenimiento.
F208.E_MAINTENANCE

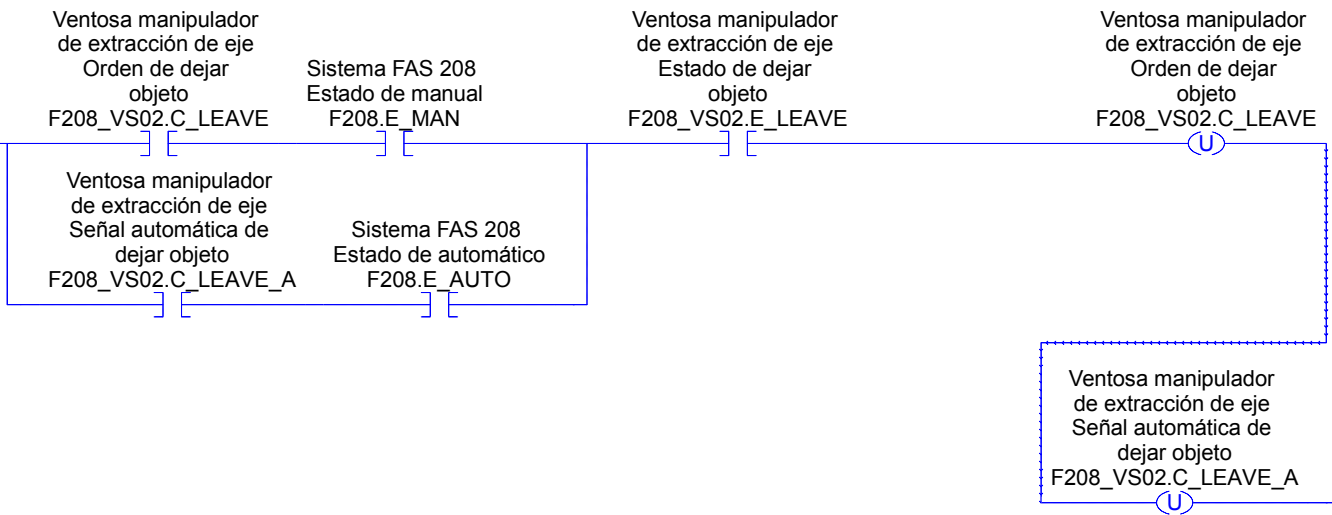
Ventosa brazo
inserción eje Señal
de salida de dejar
objeto
F208_VS01.YLEAVE

COMPORTAMIENTO DE DEJAR OBJETO FAS 208

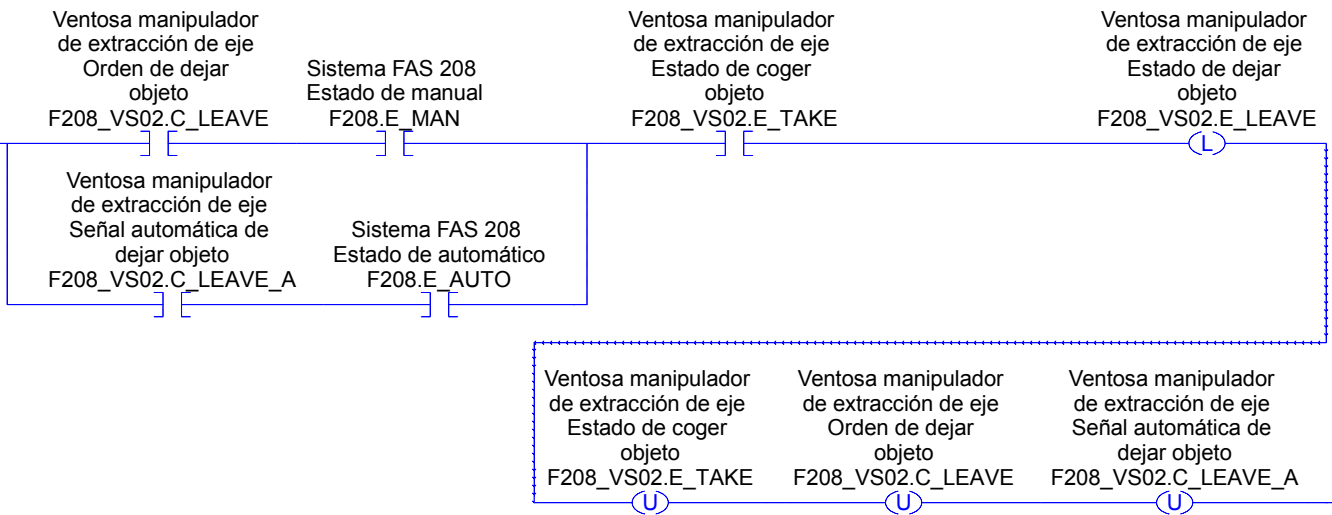
22

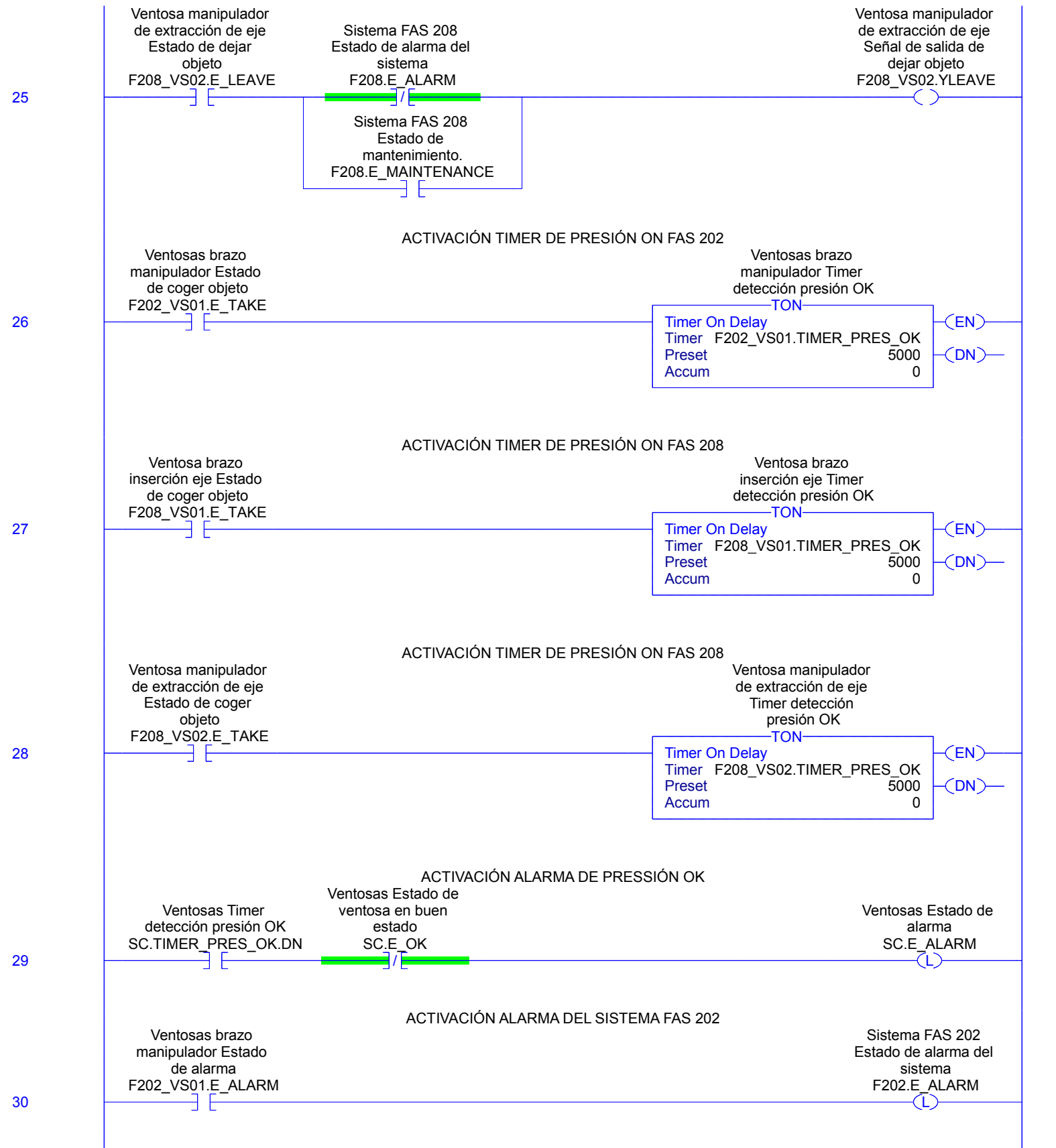


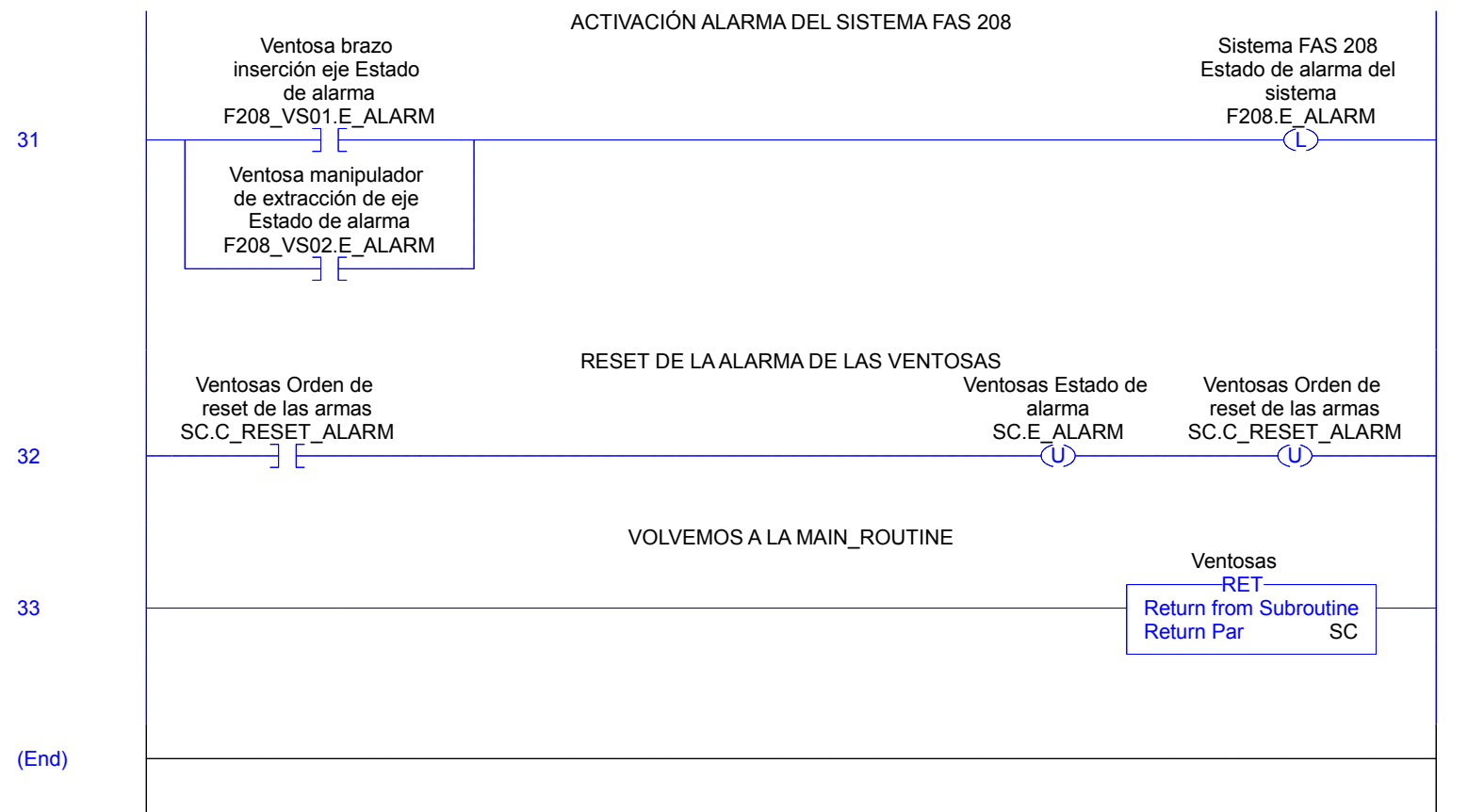
23

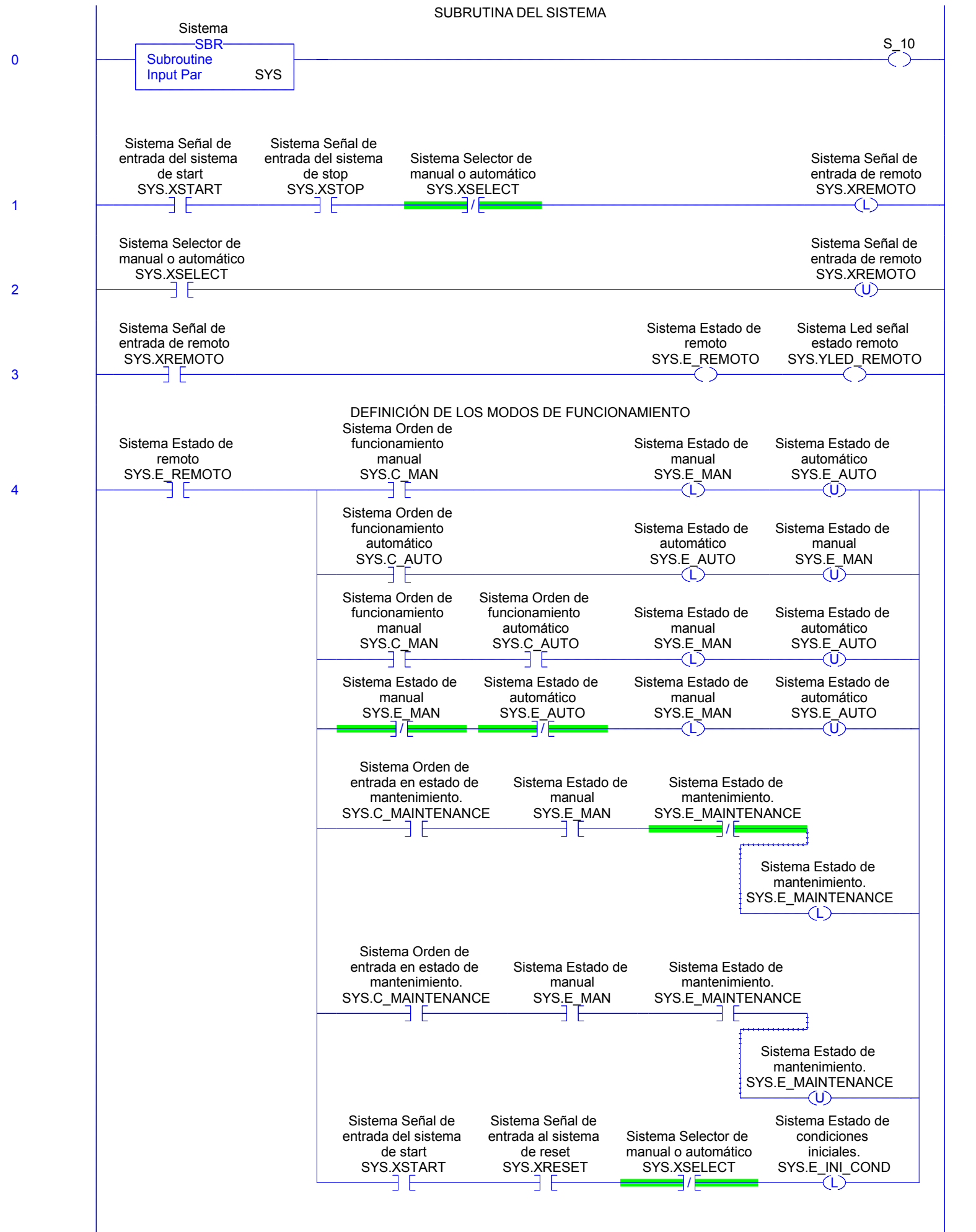


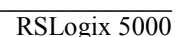
24

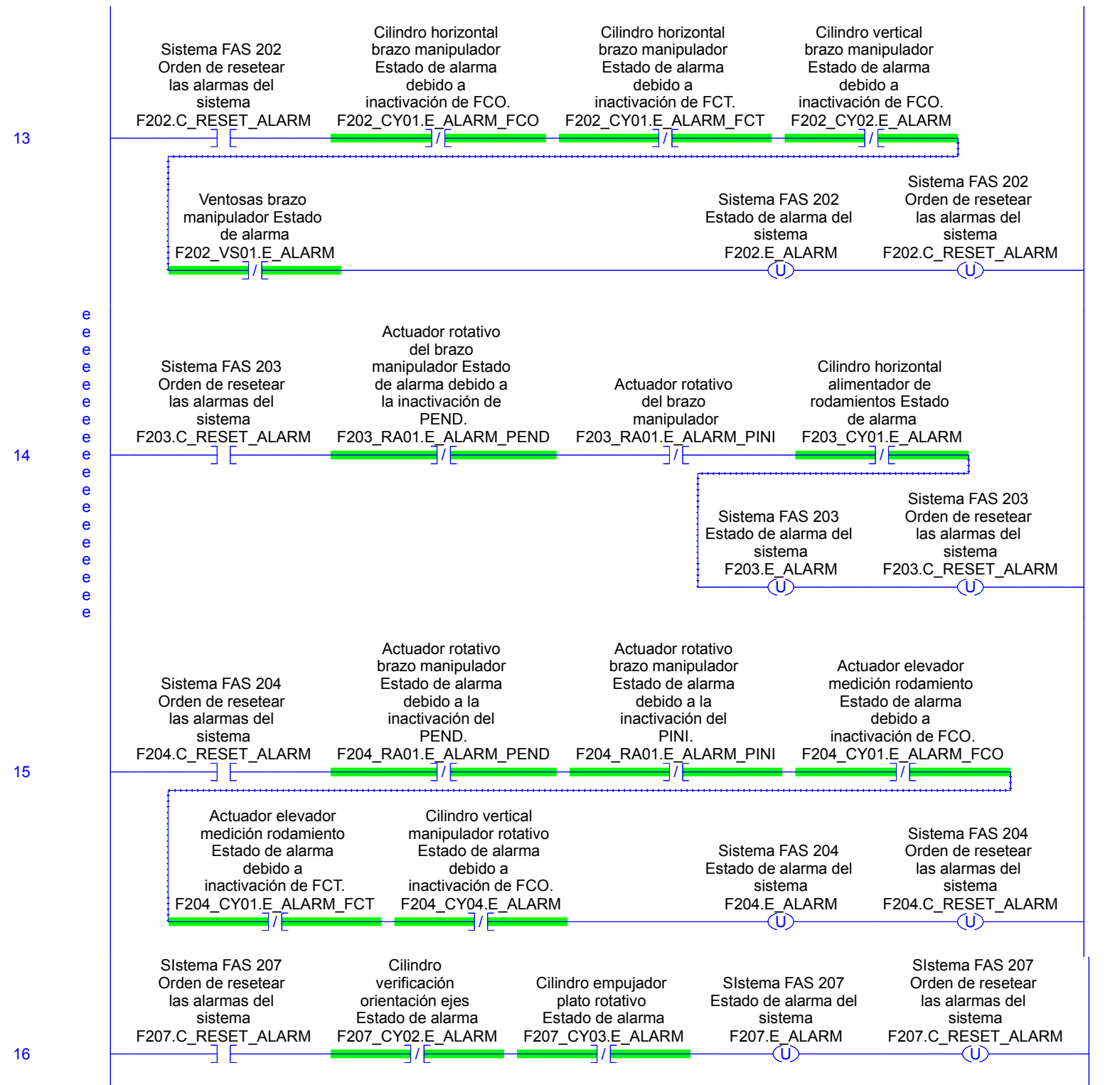


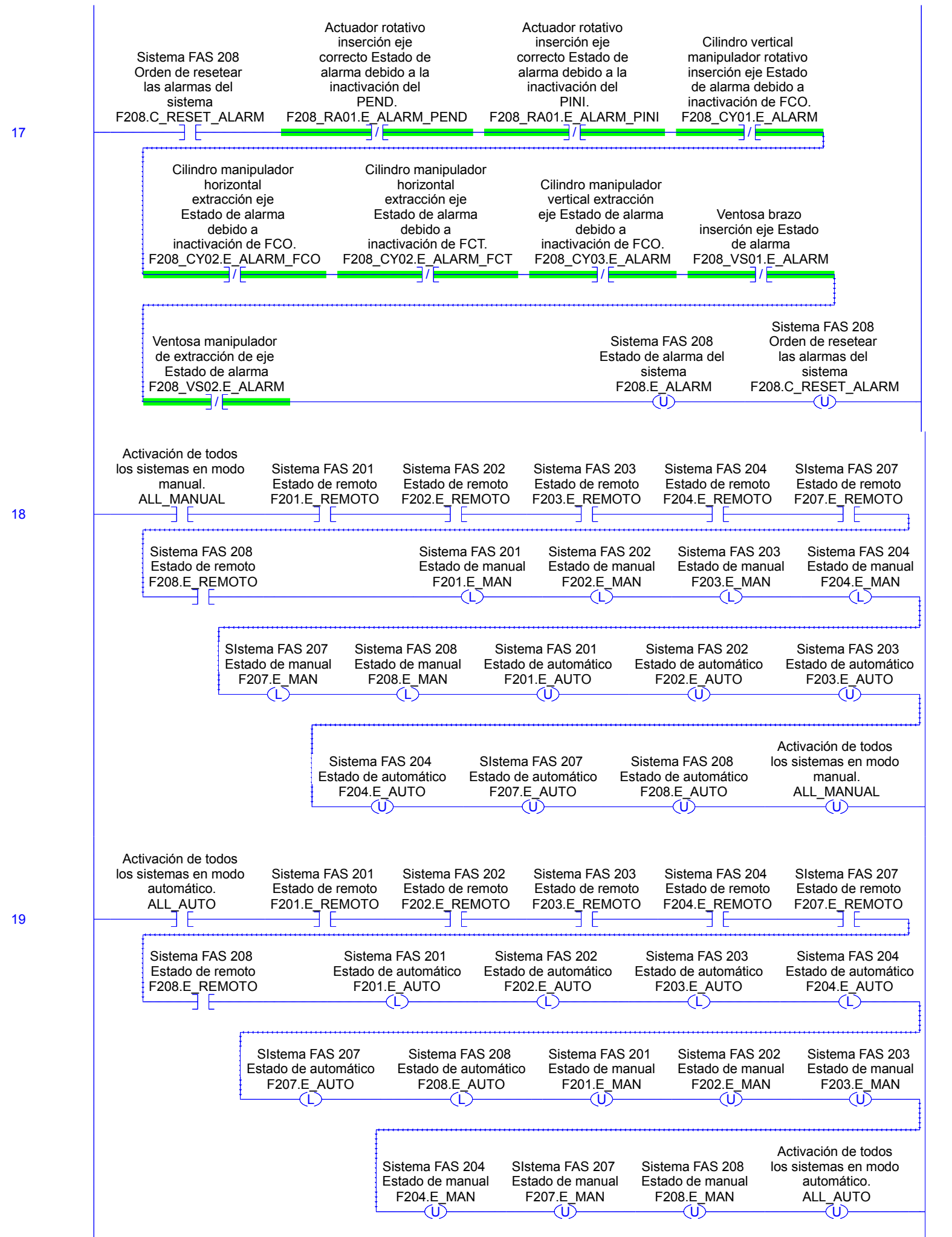


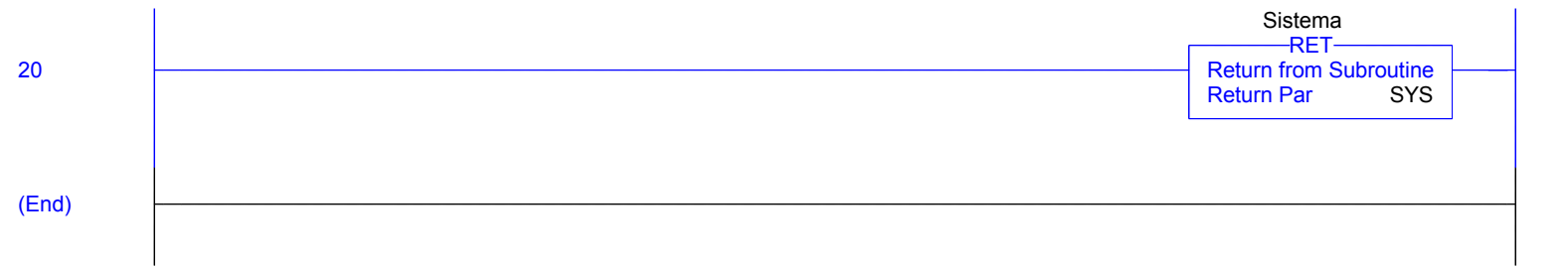


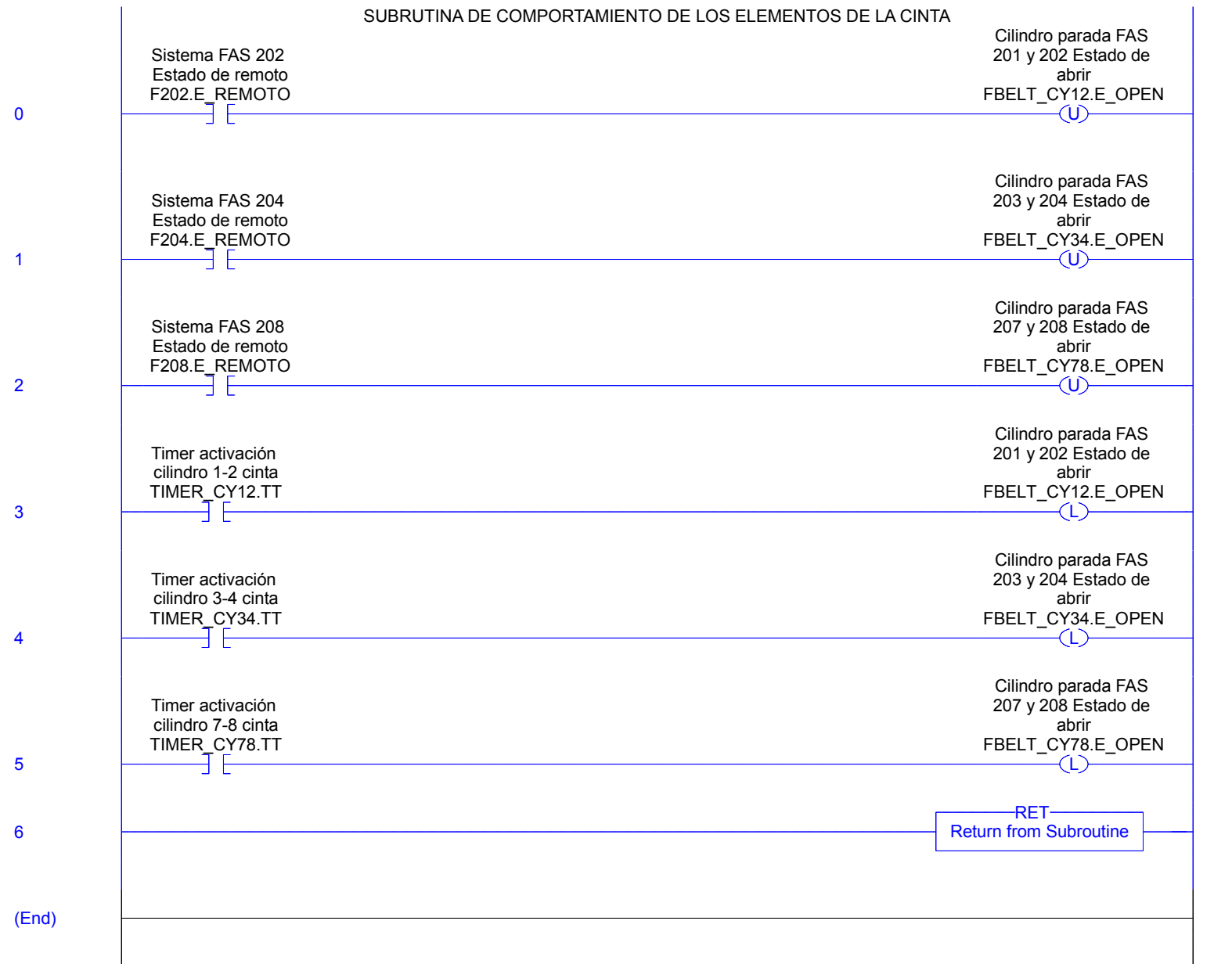




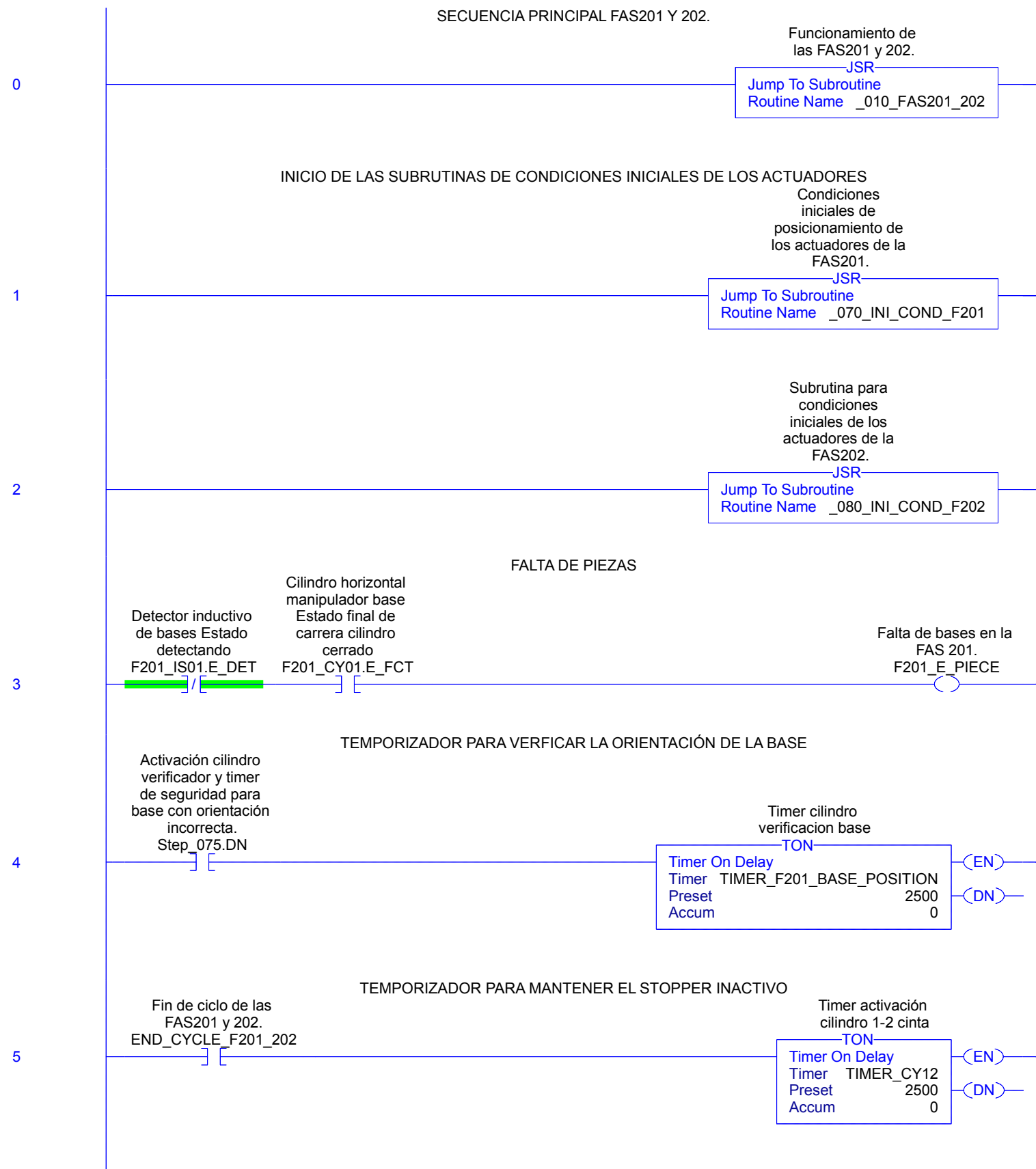


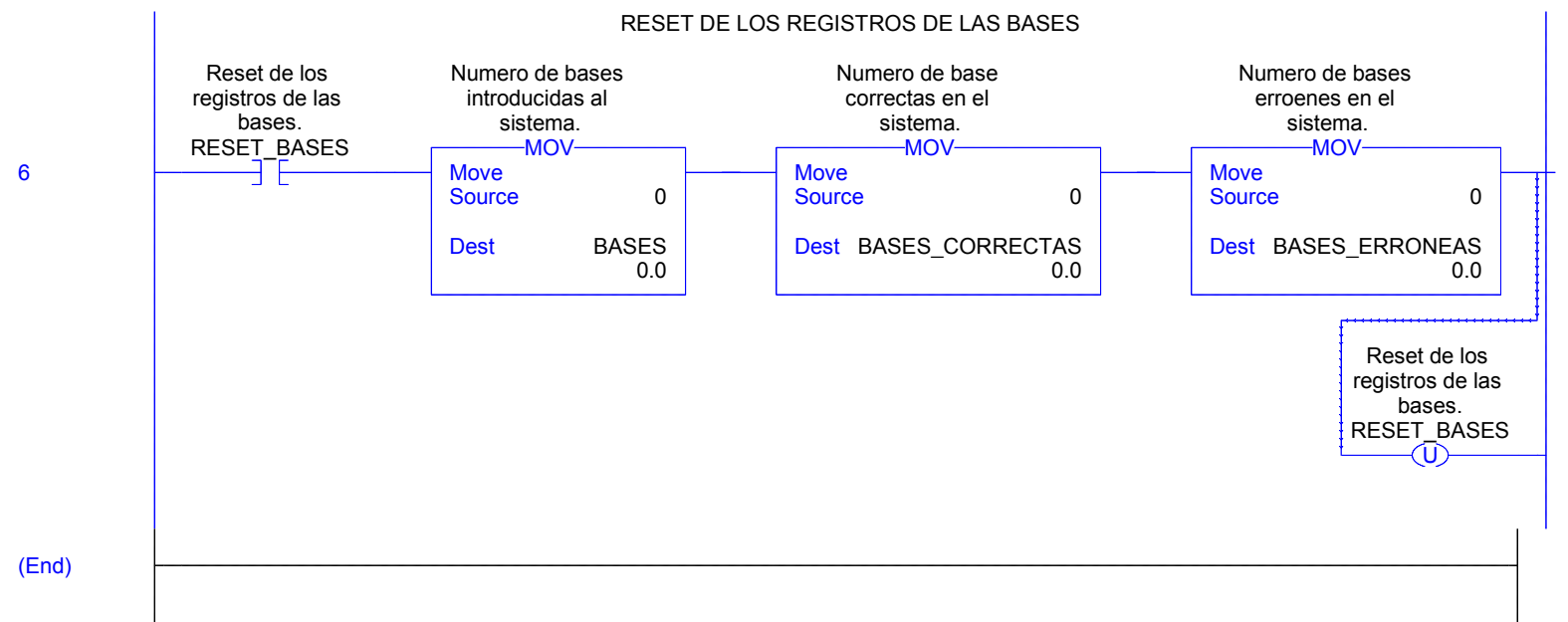




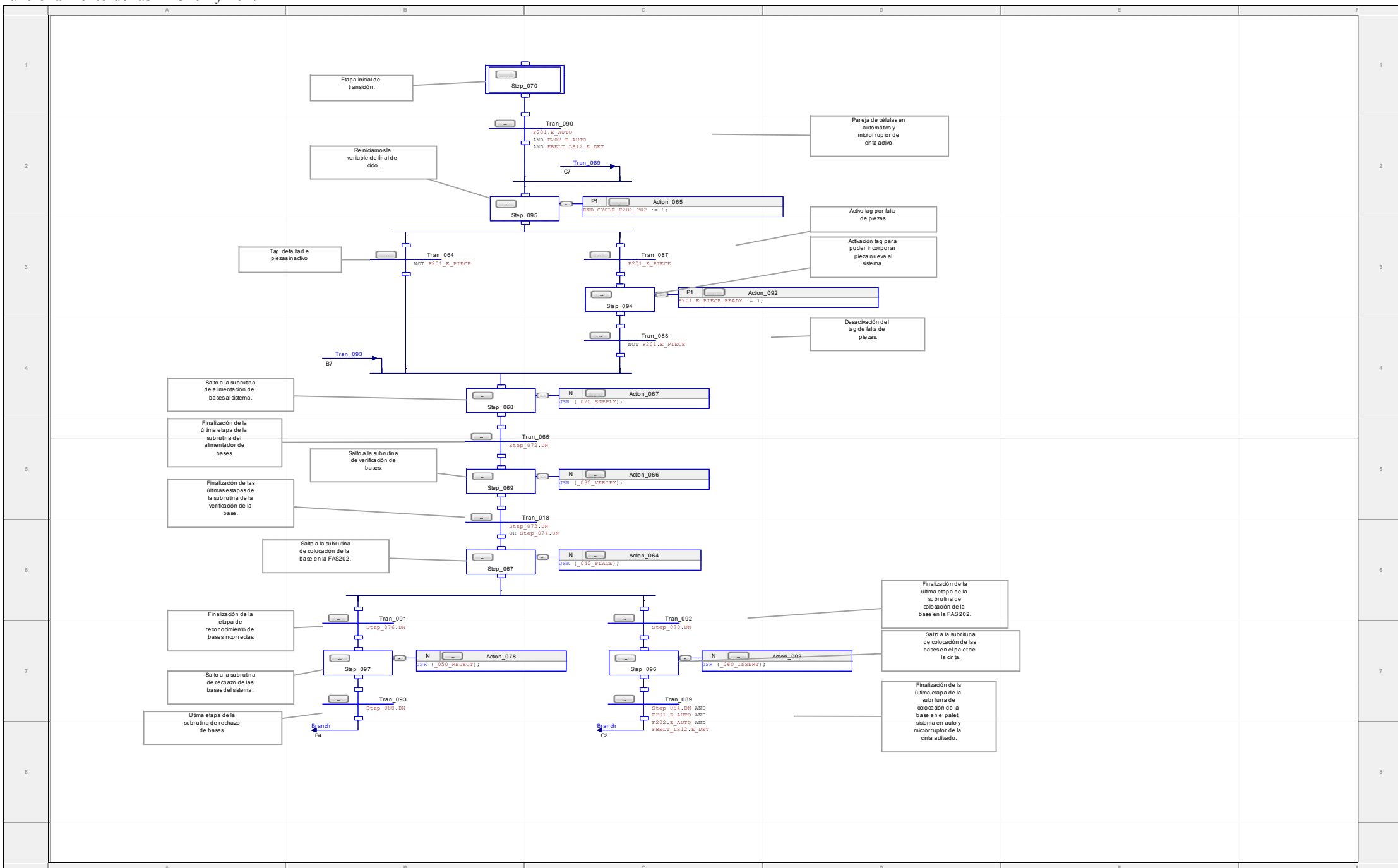


PROGRAMACIÓN FAS 201 - 202
RSLogix 5000

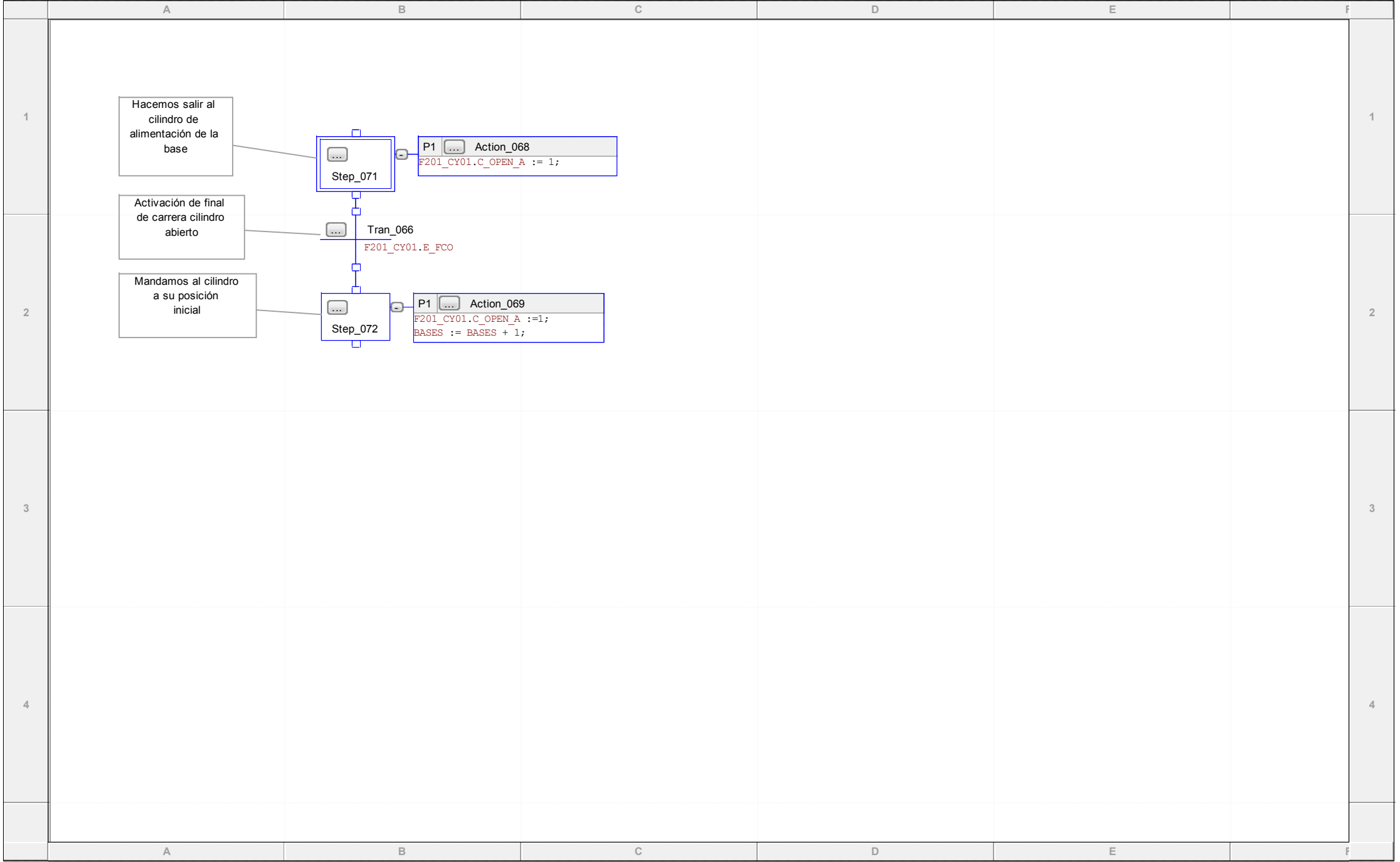




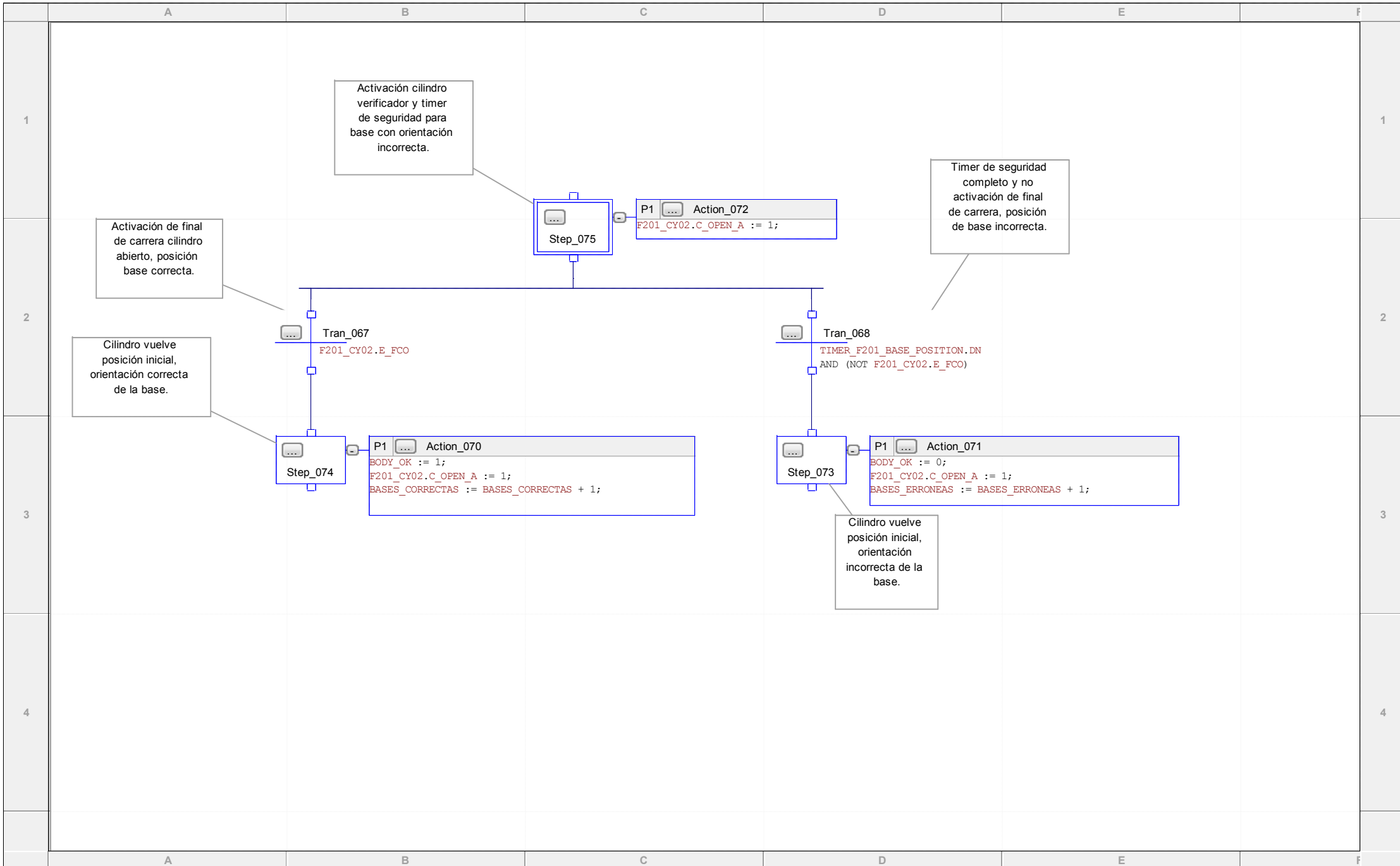
Funcionamiento de las FAS201 y 202.



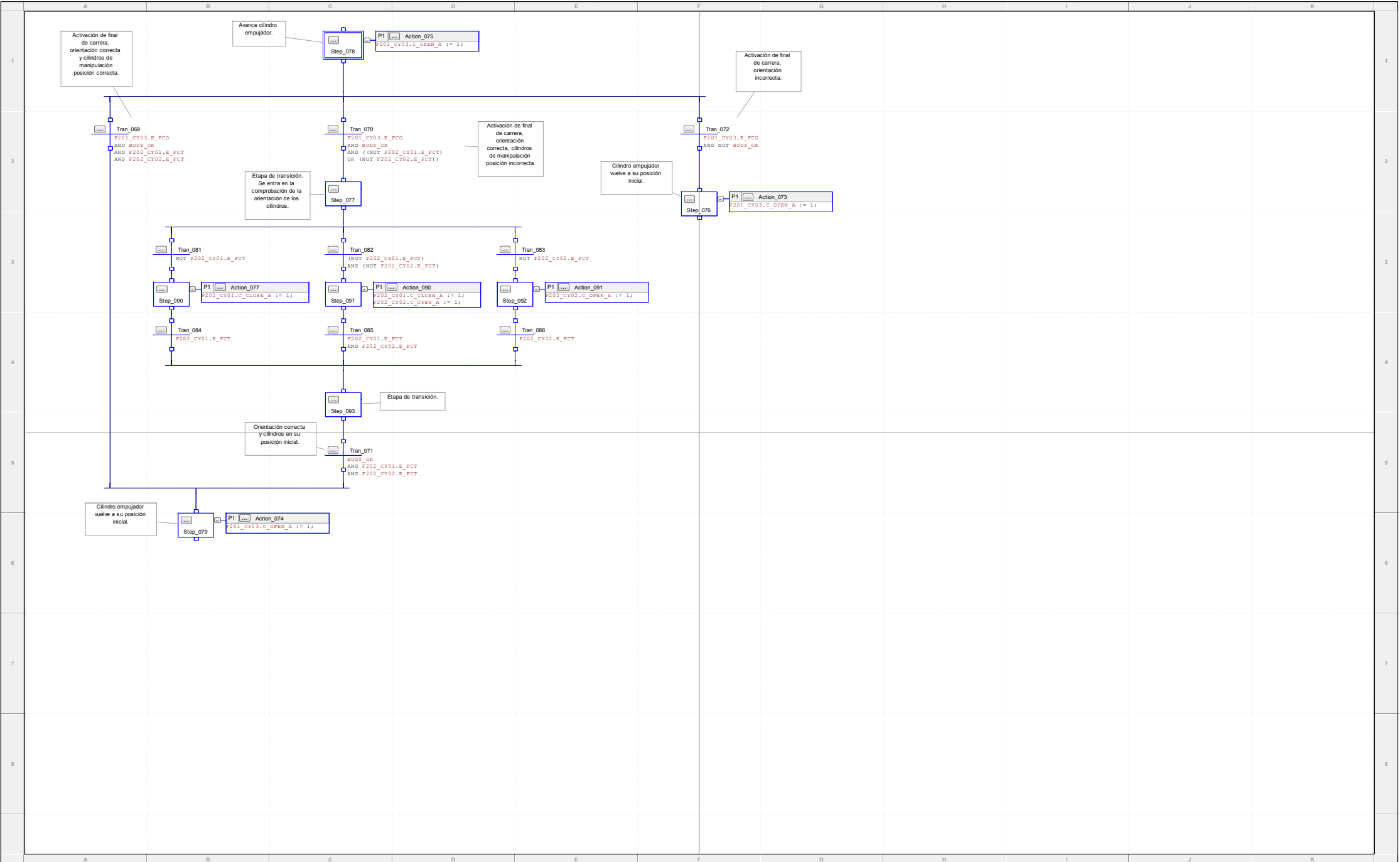
Alimentación de bases al sistema.



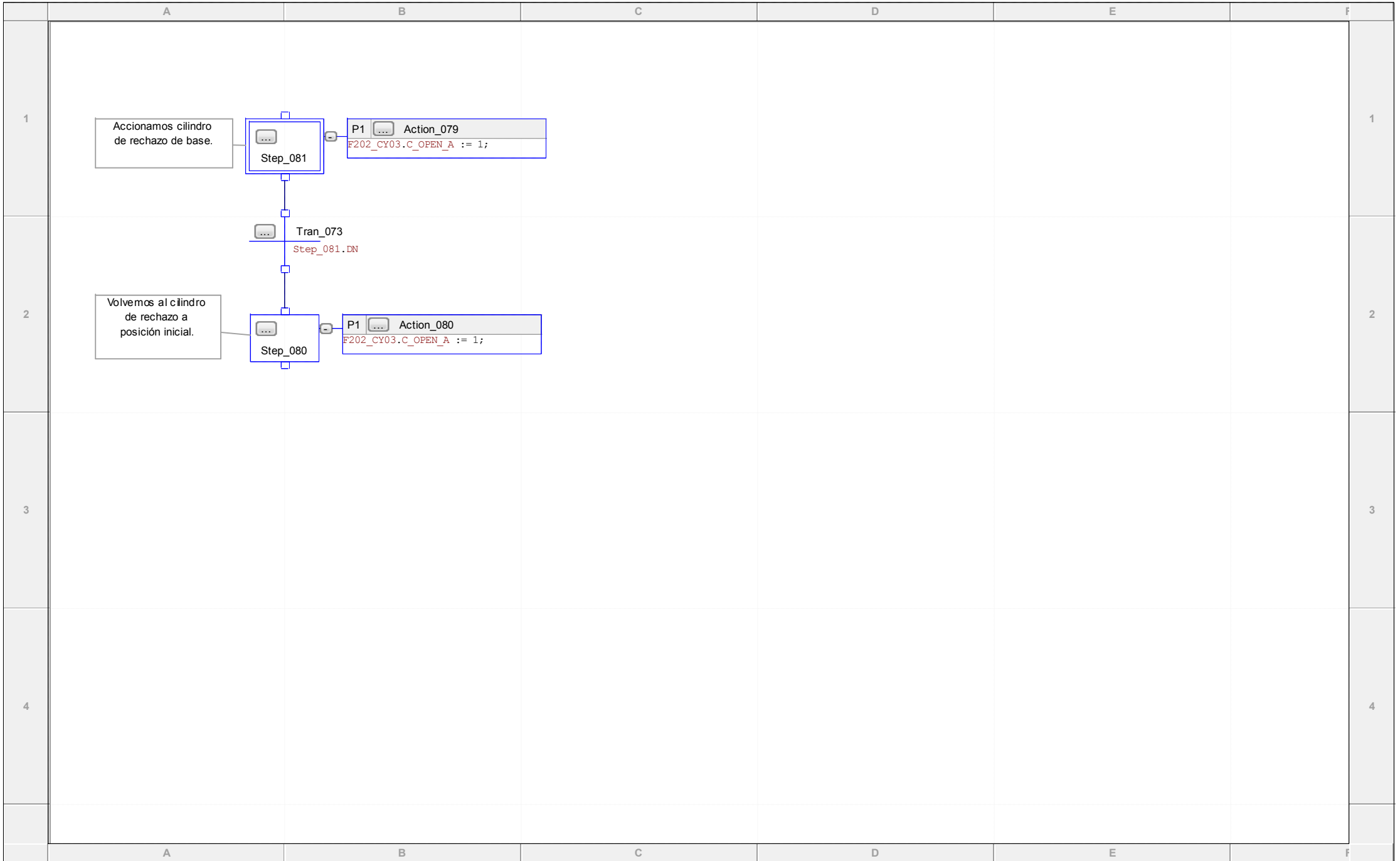
Verificación de la orientación de la base.



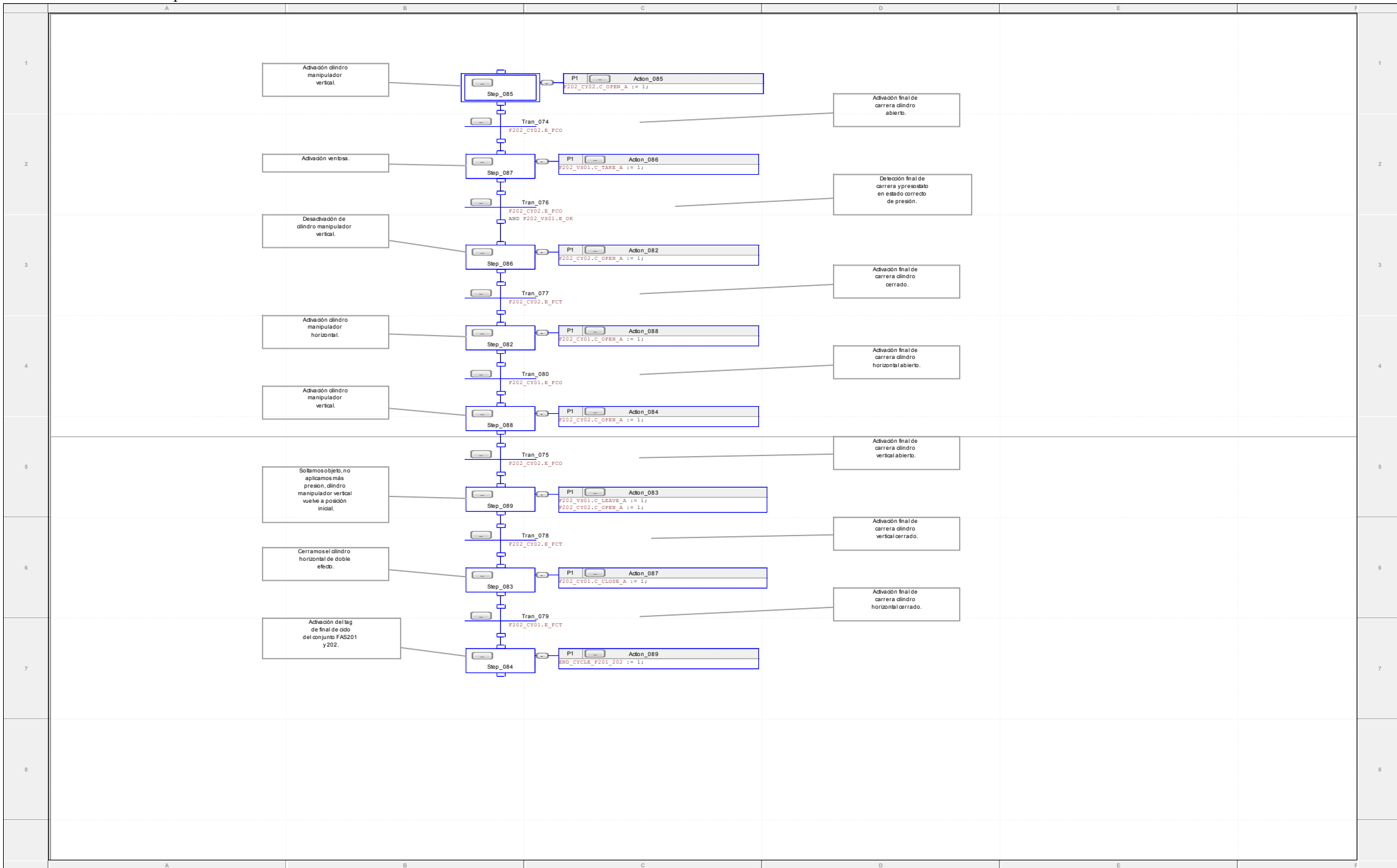
Transvase a la FAS 202.



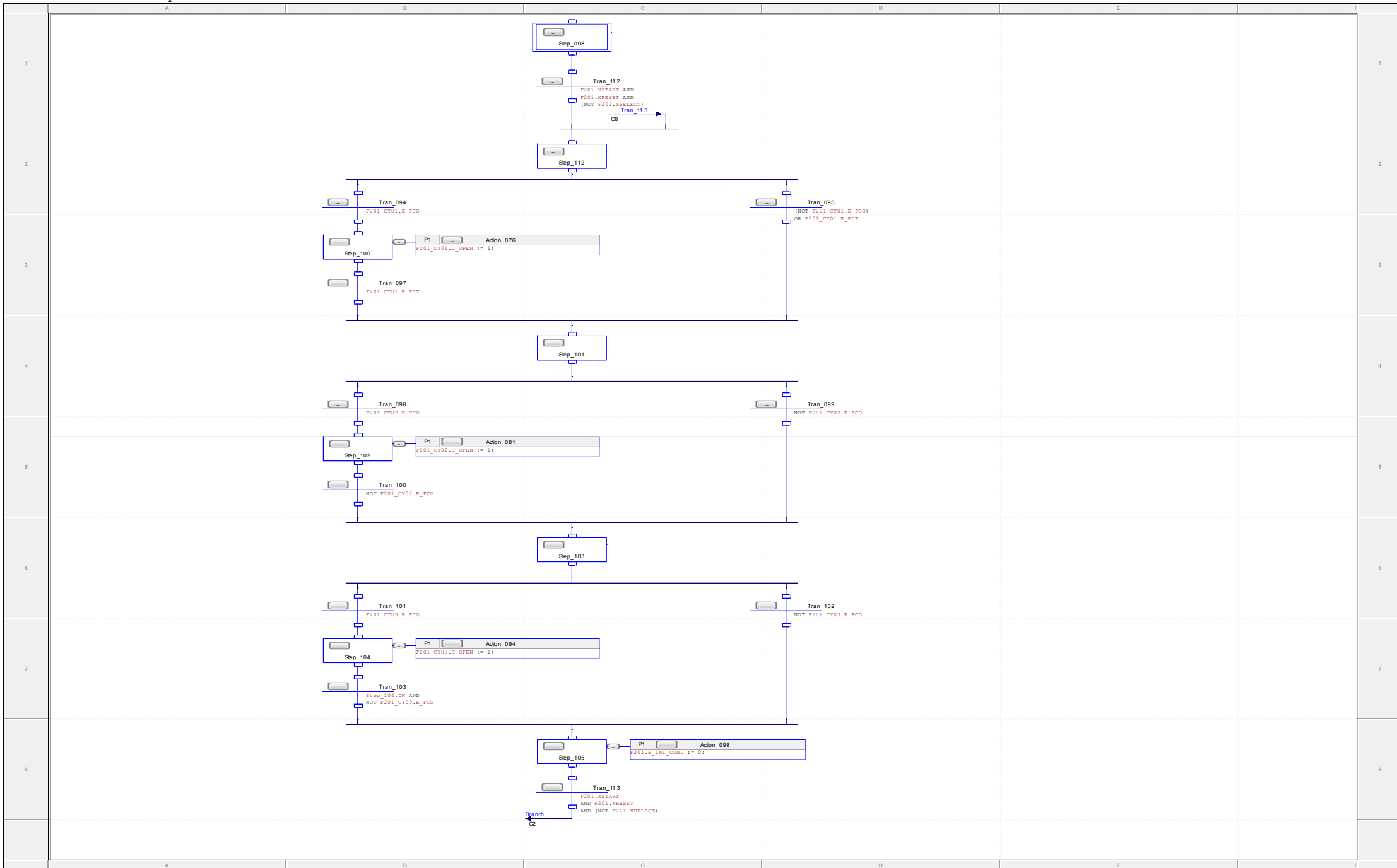
Rechazo de base con orientacion incorrecta.



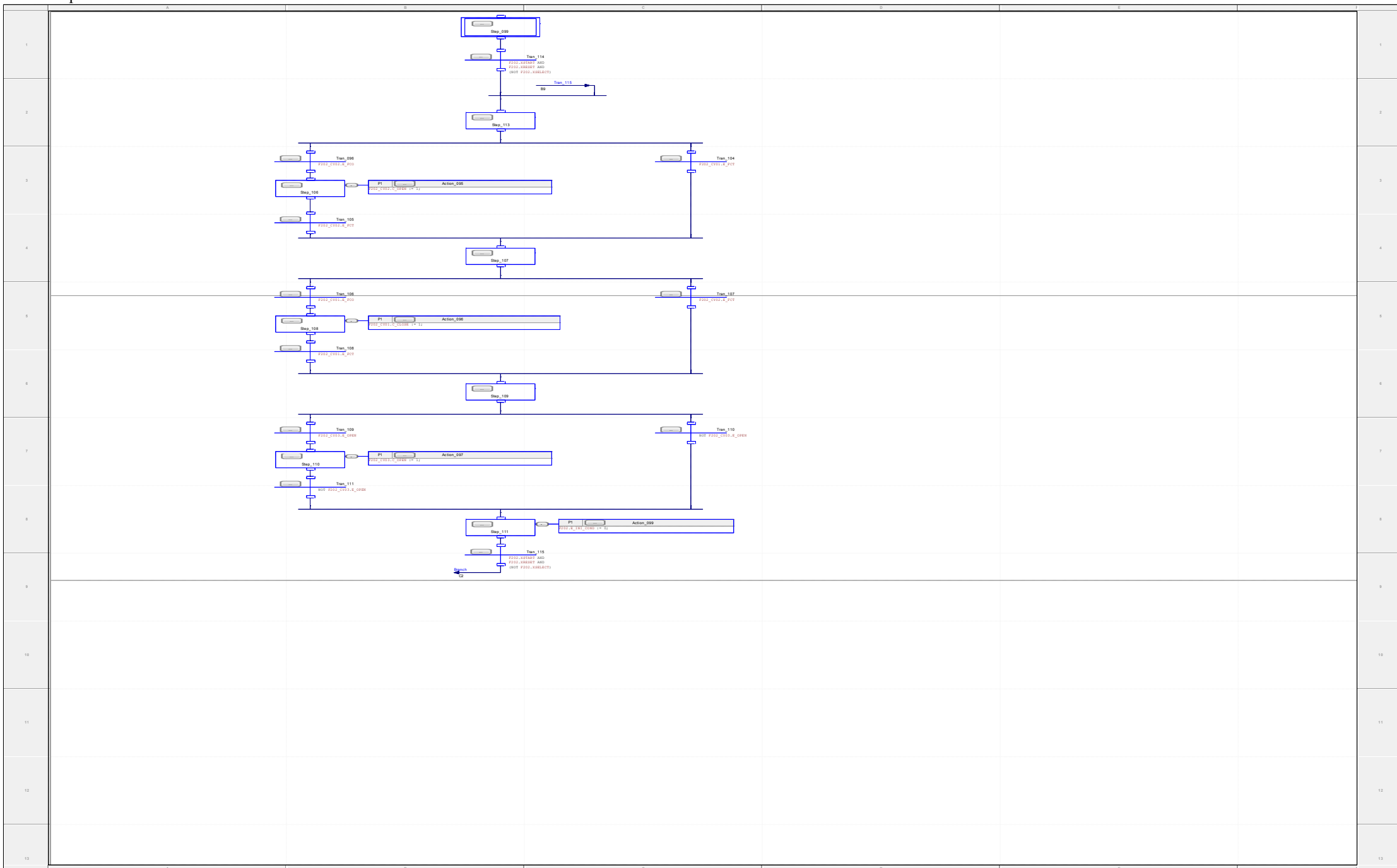
Inserción de la base en el palet de la cinta.



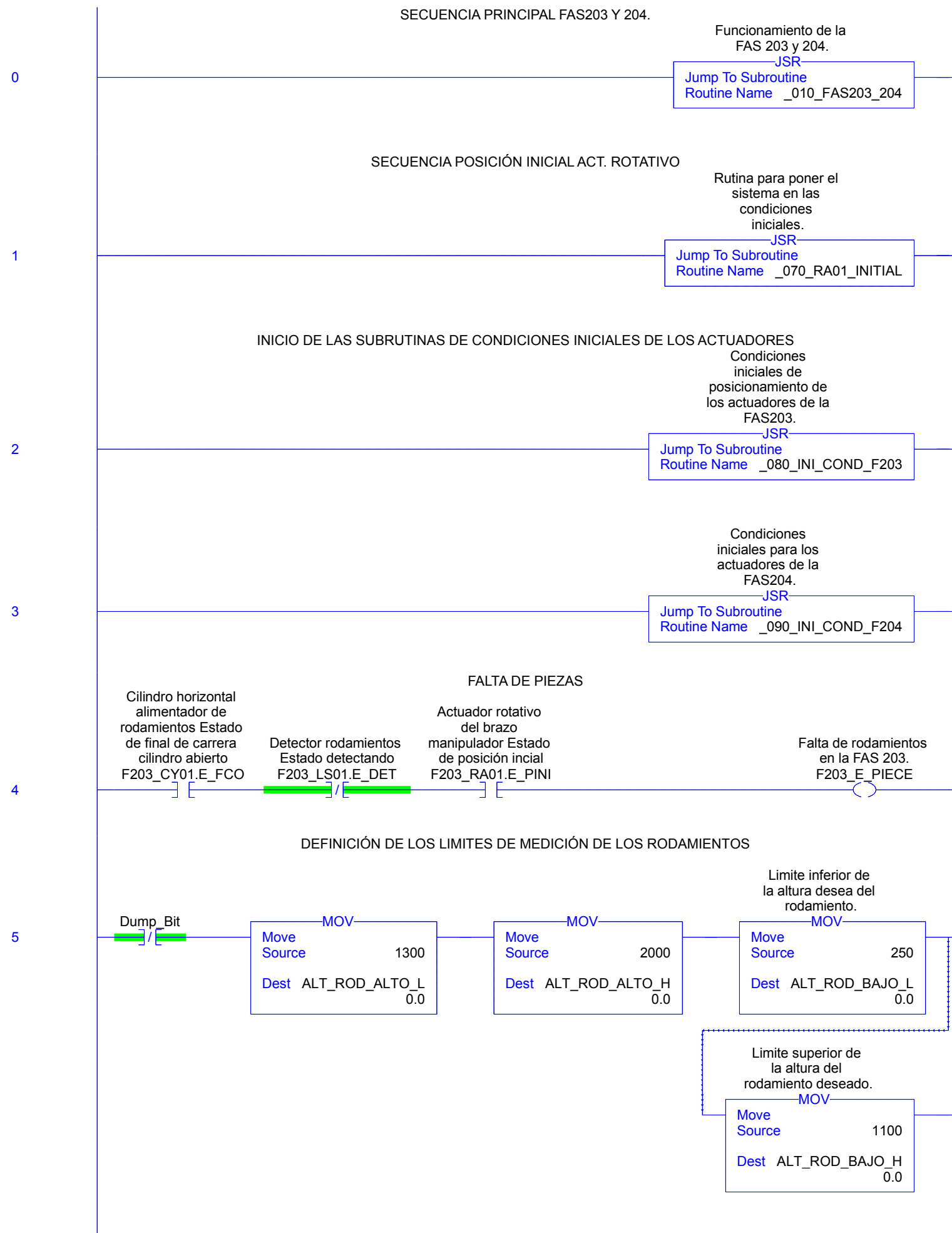
Condiciones iniciales de posicionamiento de los actuadores de la FAS201.

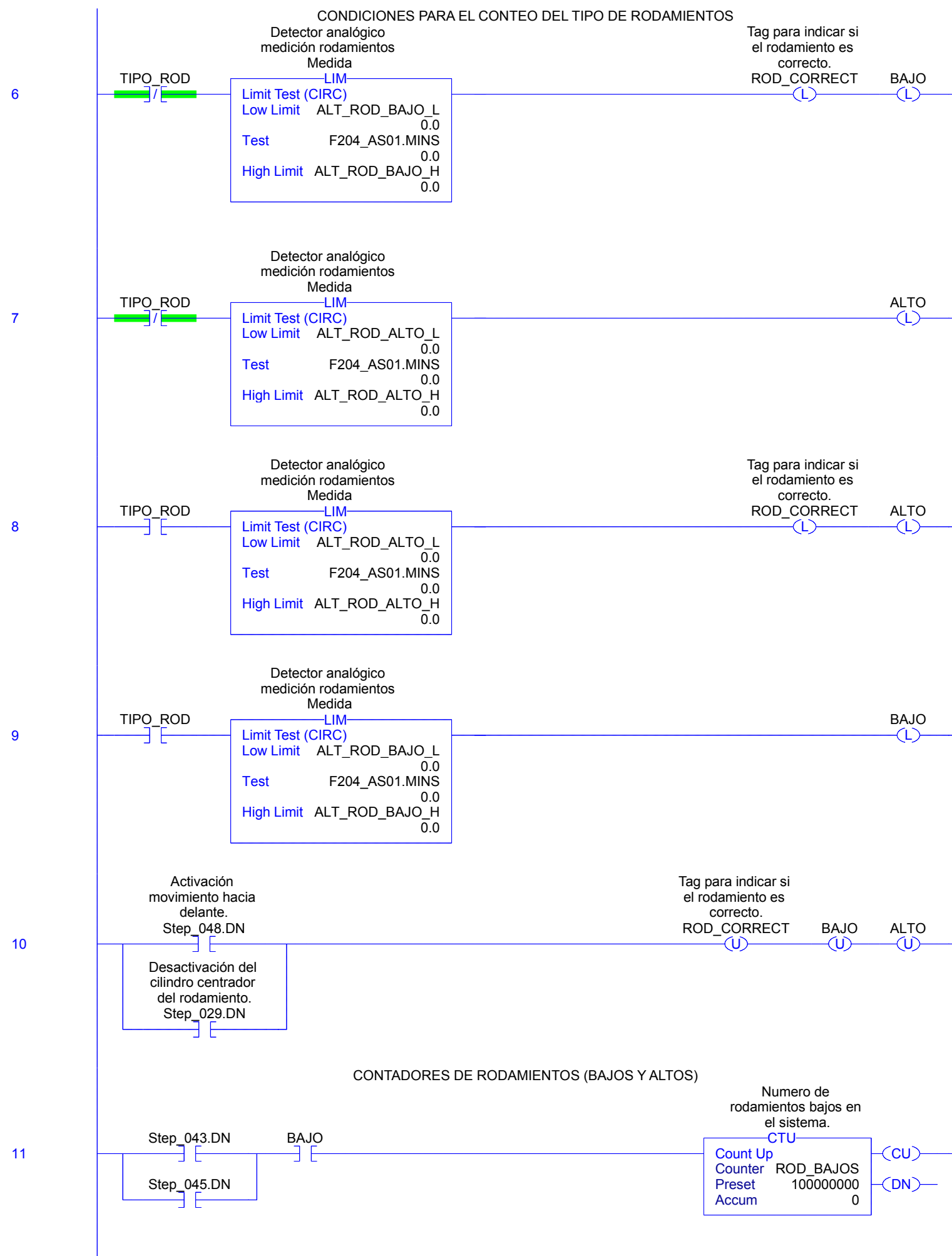


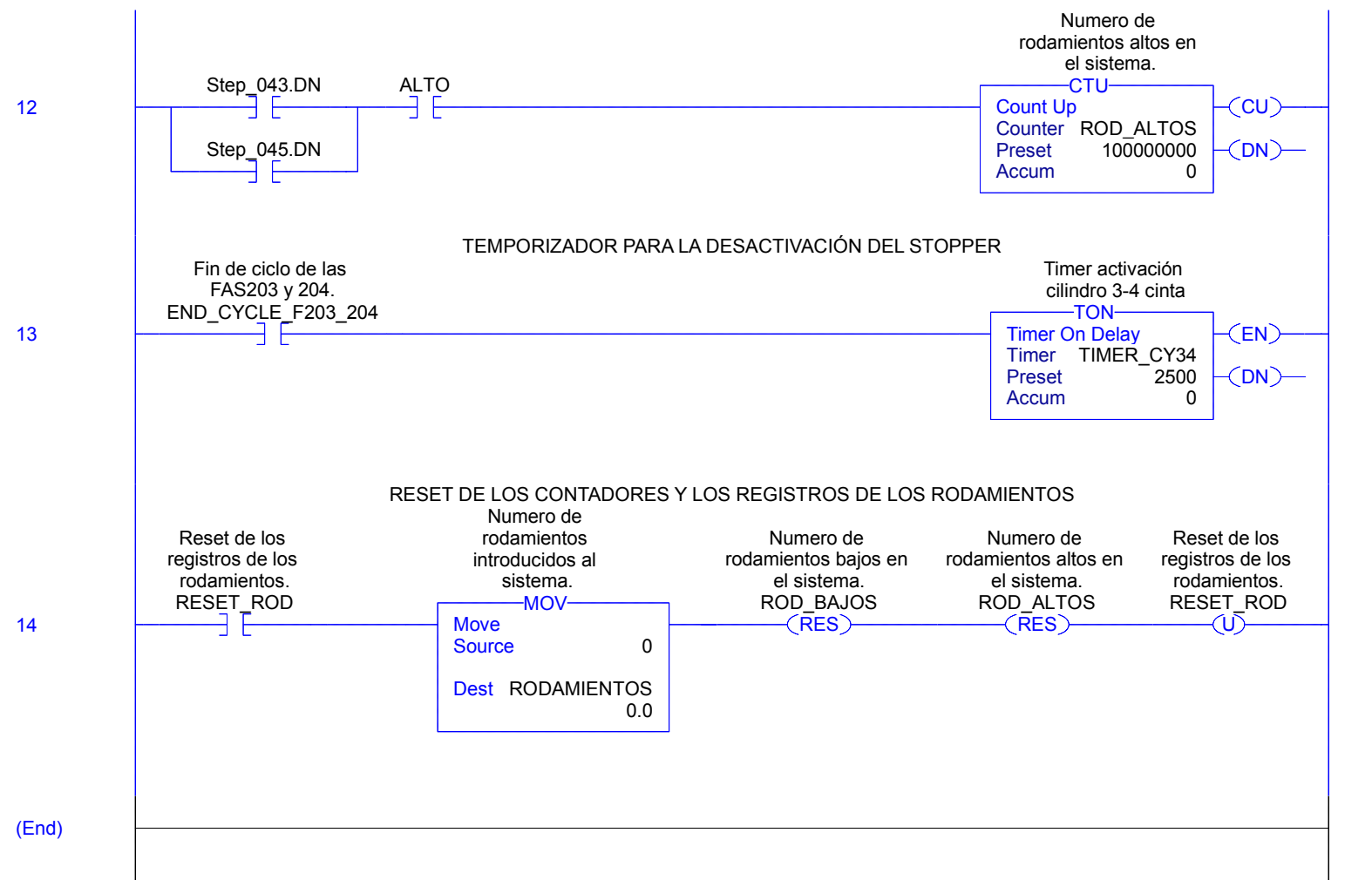
Subrutina para condiciones iniciales de los actuadores de la FAS202.



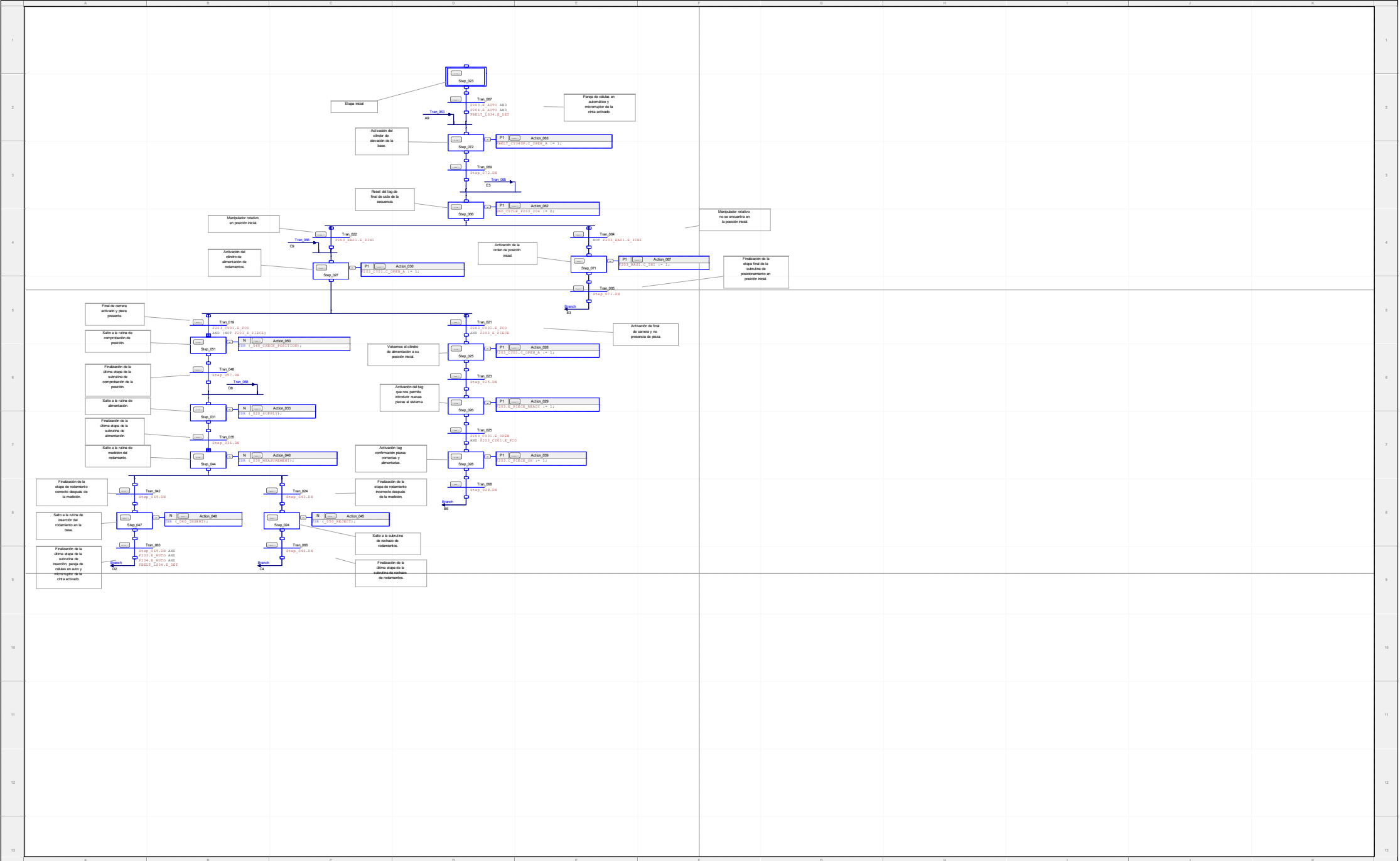
PROGRAMACIÓN FAS 203 - 204
RSLogix 5000



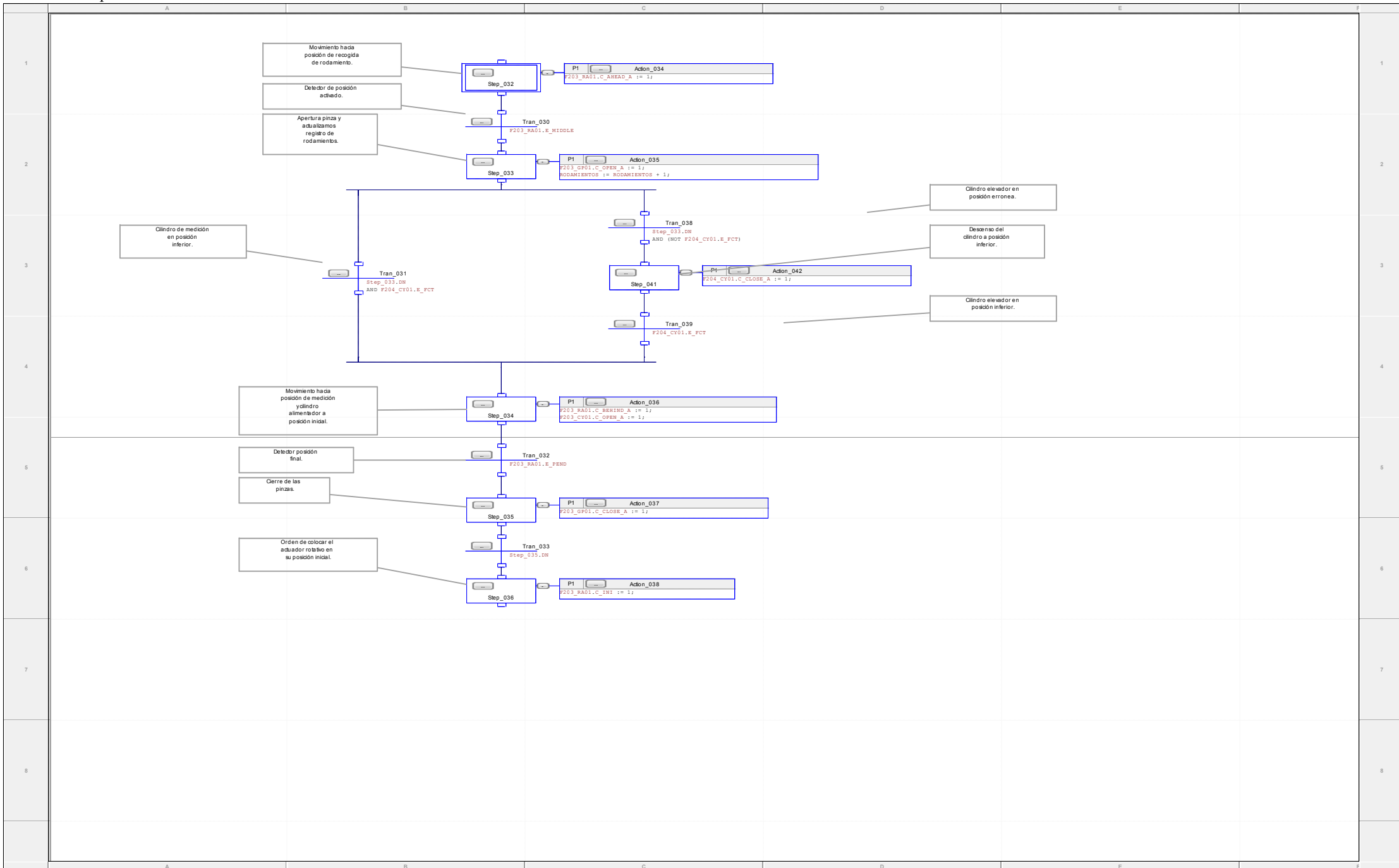




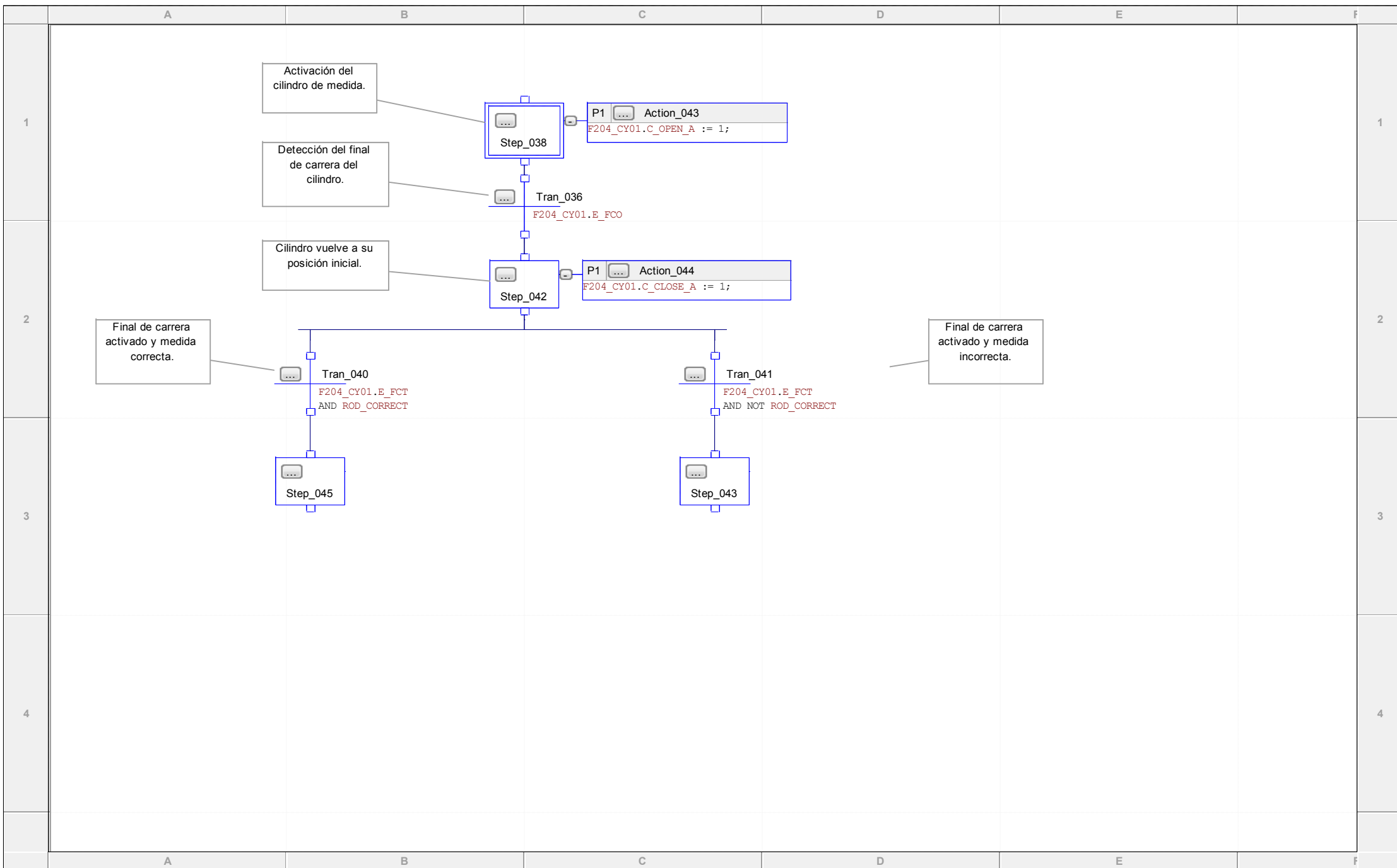
Funcionamiento de la FAS 203 y 204.



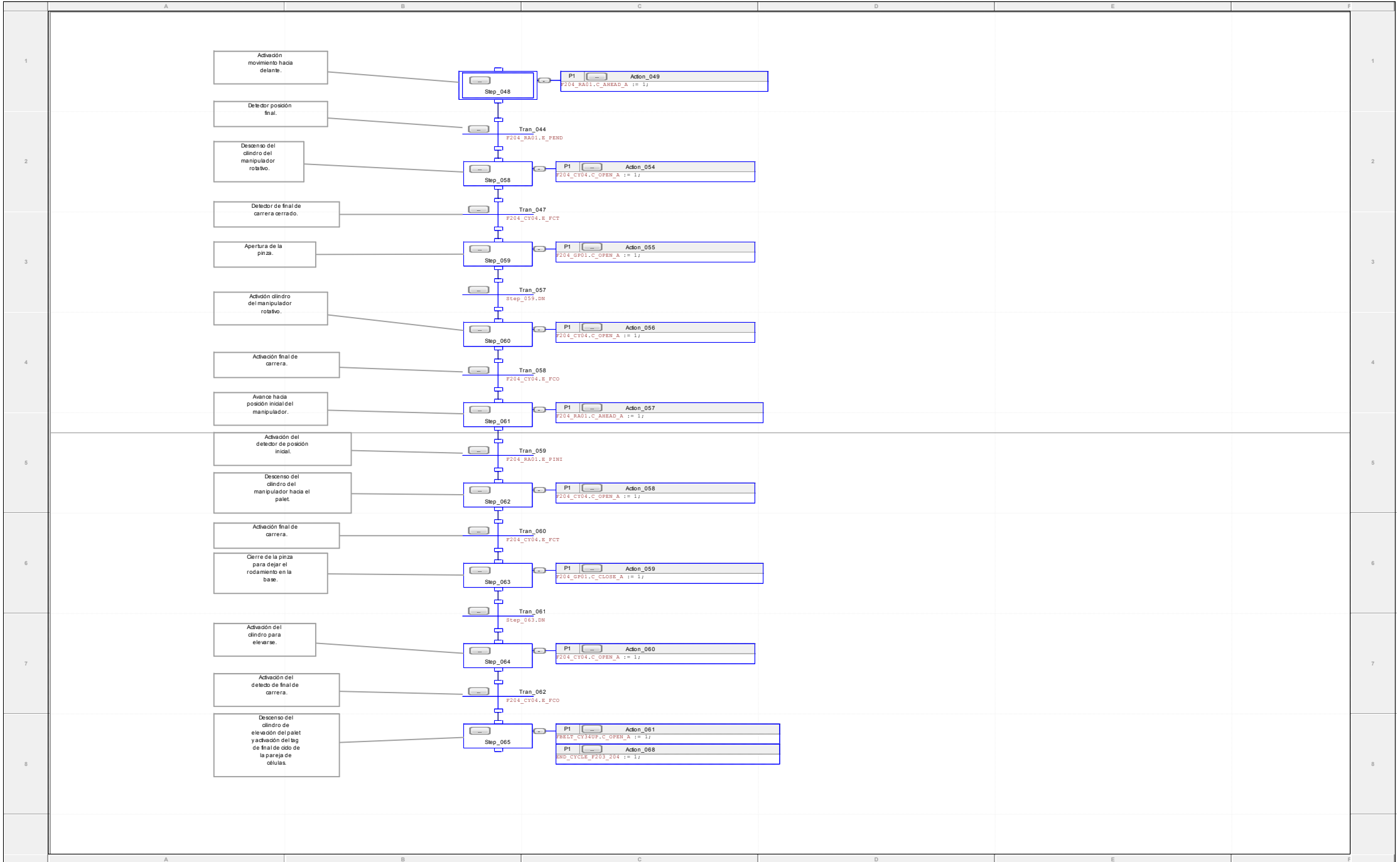
Transvase a la posición de medición del rodamiento en la FAS204.



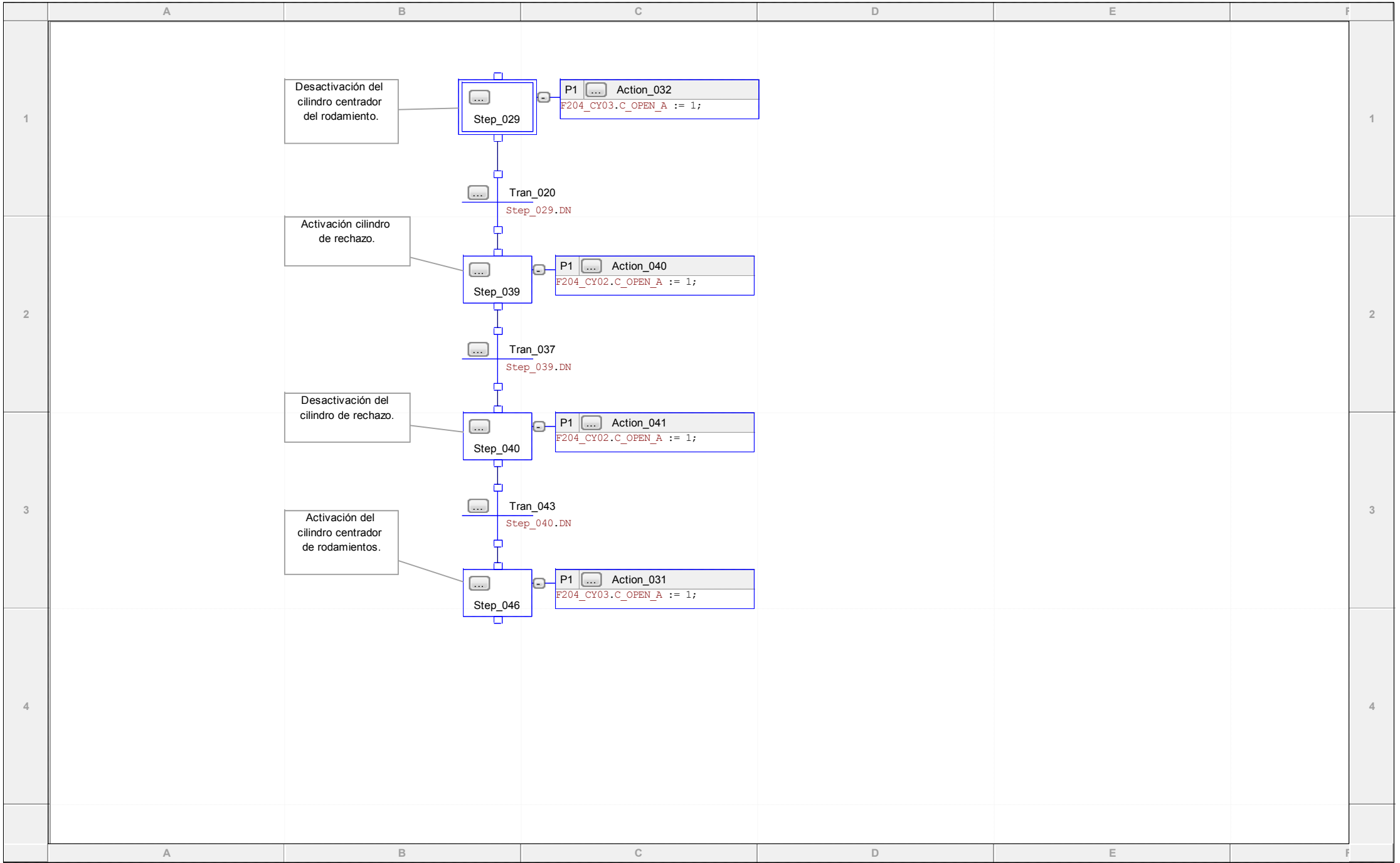
Medición de la altura del rodamiento.



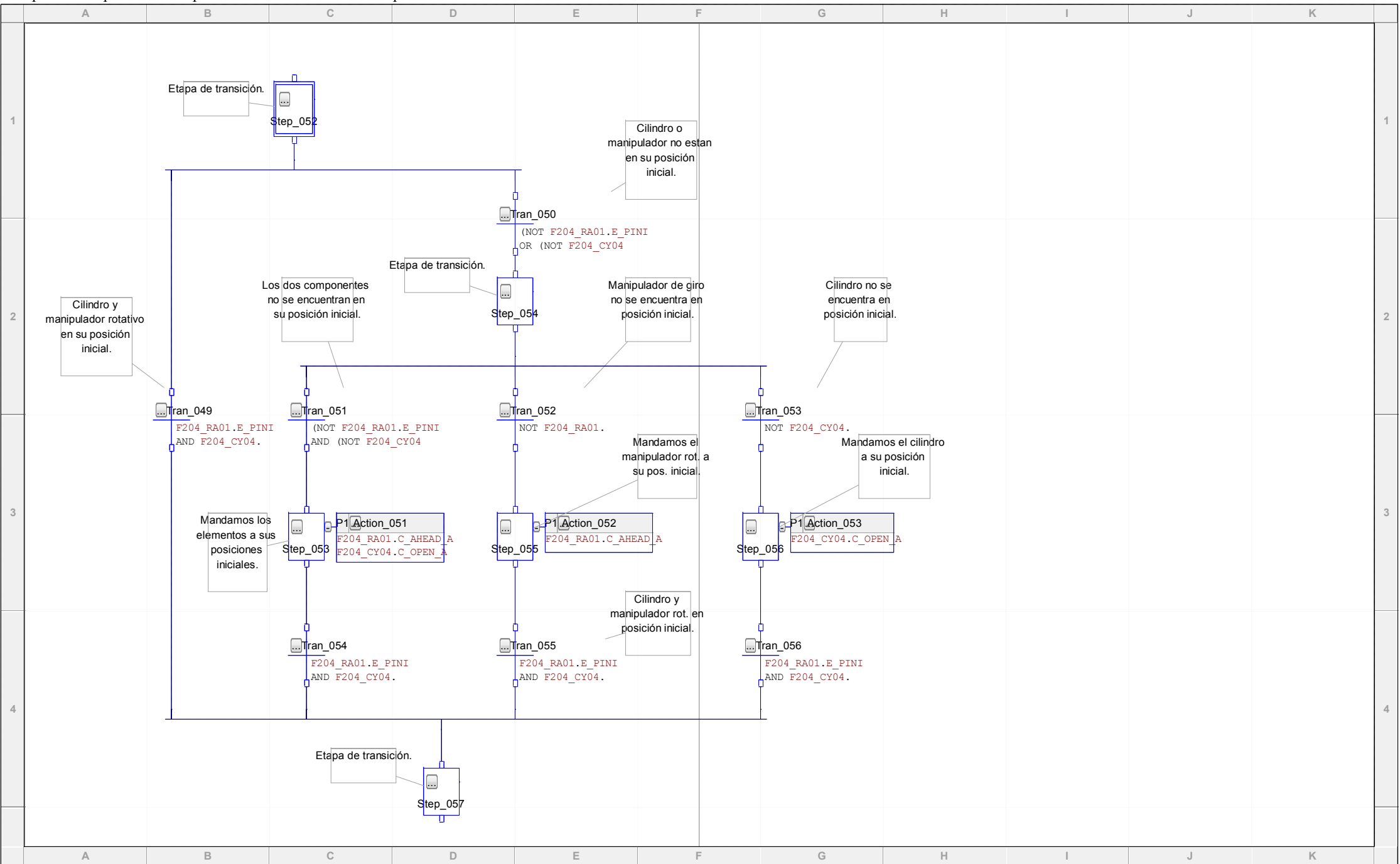
Inserción del rodamiento en la base.



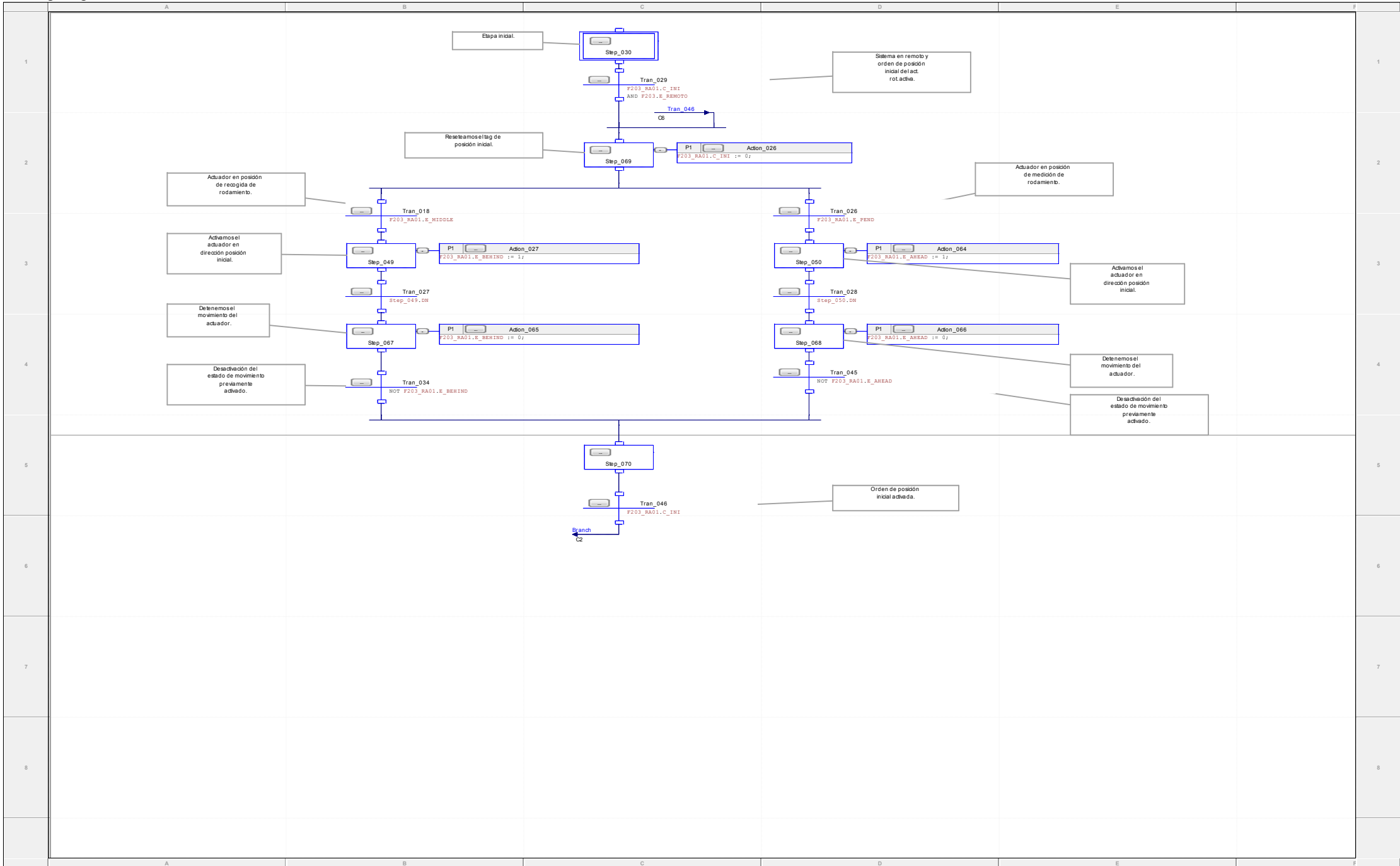
Rechazo de rodamiento incorrecto.



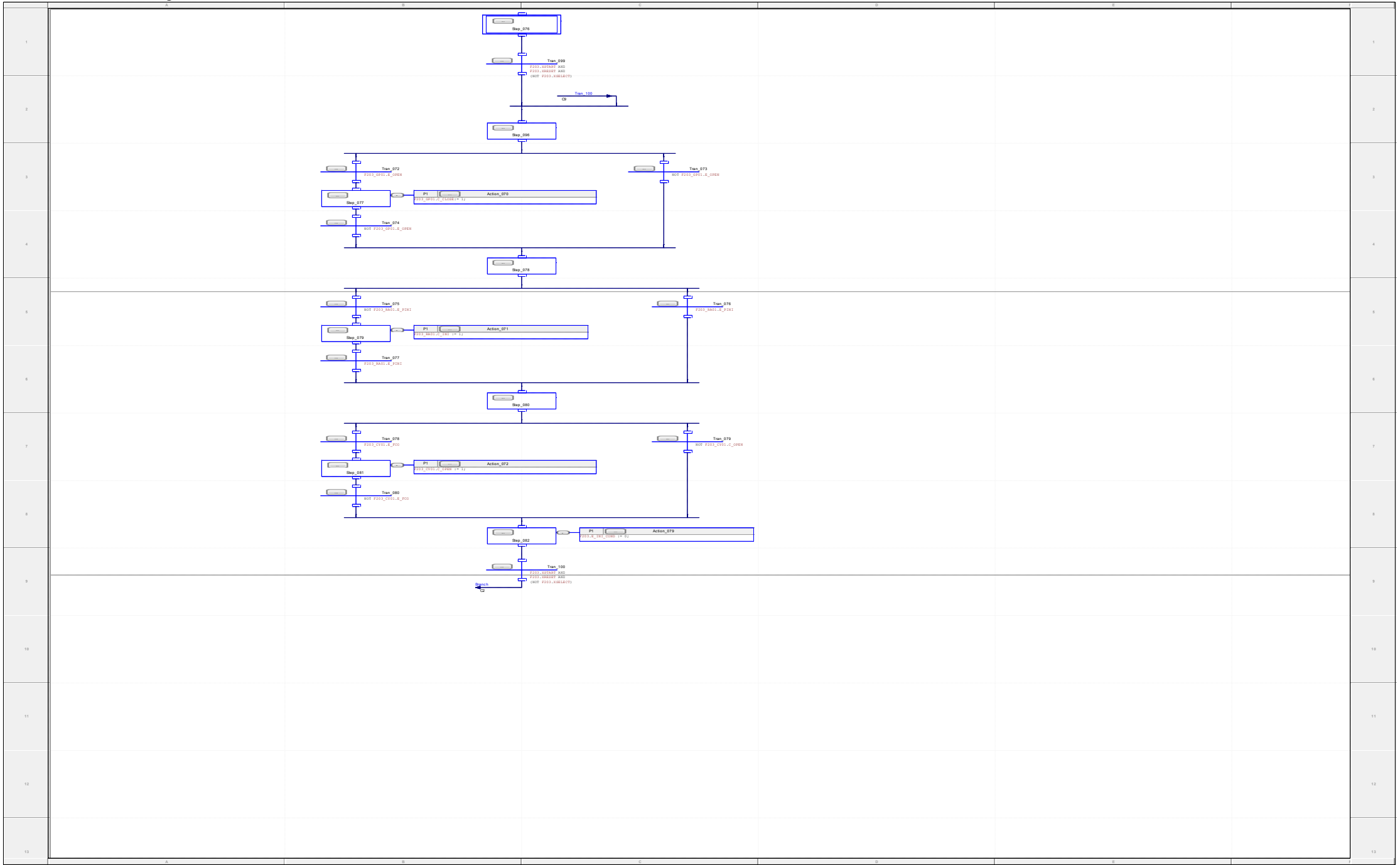
Comprobación posición componentes FAS204 en correcta posición.



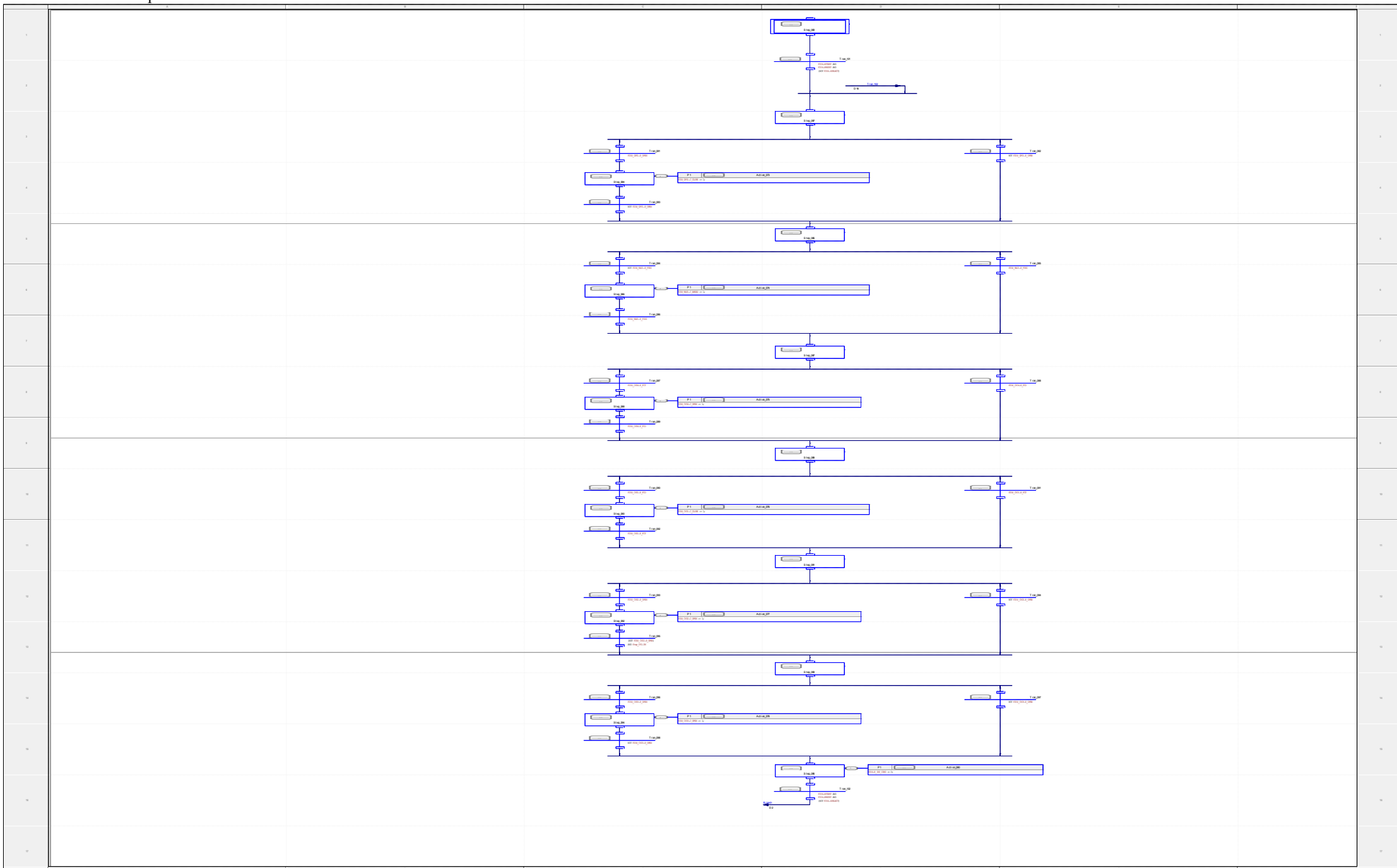
Rutina para poner el sistema en las condiciones iniciales.



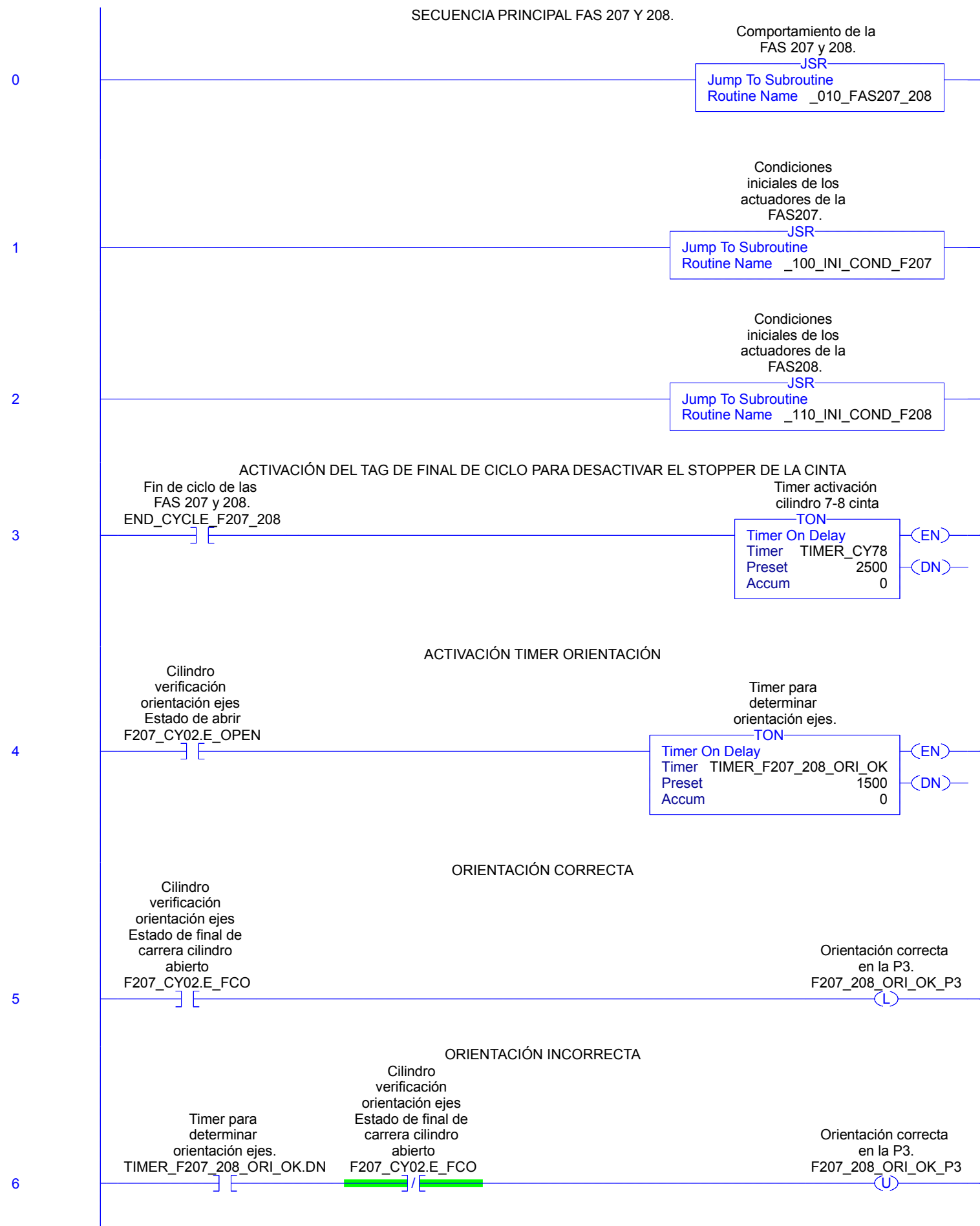
Condiciones iniciales de posicionamiento de los actuadores de la FAS203.

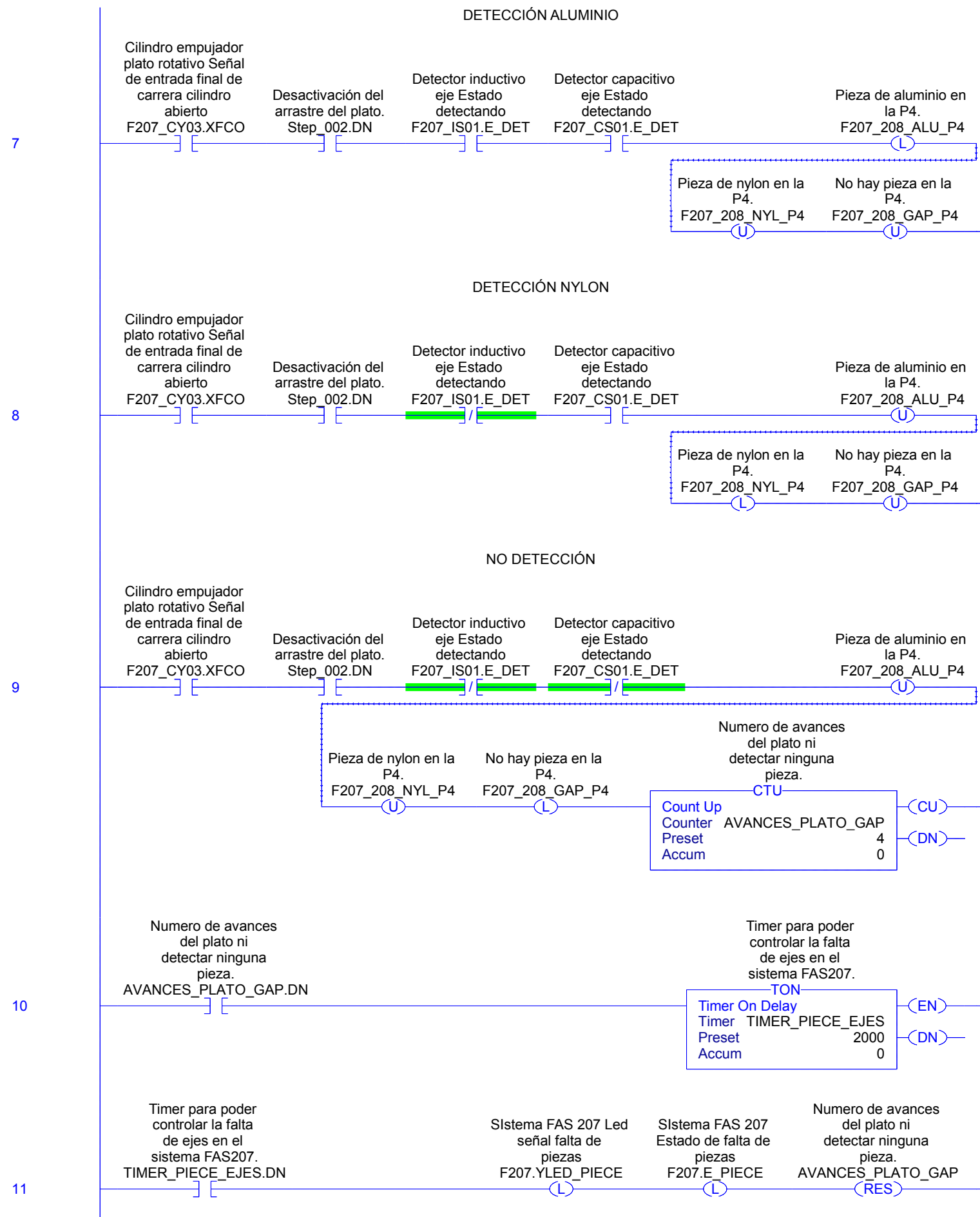


Condiciones iniciales para los actuadores de la FAS204.



PROGRAMACIÓN FAS 207 - 208
RSLogix 5000





12

ACTUALIZACIÓN REGISTROS P5 A P6 (POSICIONES)

Cilindro empujador
plato rotativo
Estado de abrir
F207_CY03.E_OPEN

Tag para poder
habilitar los
registros.
F207_208_UNABLE_REGISTERS

Instr. ONS para
registro de
posición.
F207_208_ONS3_REG
[ONS]

Pieza de aluminio en
la P5.
F207_208_ALU_P5

Pieza de aluminio en
la P6.
F207_208_ALU_P6

Pieza de aluminio en
la P5.
F207_208_ALU_P5

Pieza de aluminio en
la P6.
F207_208_ALU_P6

Pieza de nylon en la
P5.
F207_208_NYL_P5

Pieza de nylon en la
P6.
F207_208_NYL_P6

Pieza de nylon en la
P5.
F207_208_NYL_P5

Pieza de nylon en la
P6.
F207_208_NYL_P6

No hay pieza en la
P5.
F207_208_GAP_P5

No hay pieza en la
P6.
F207_208_GAP_P6

No hay pieza en la
P5.
F207_208_GAP_P5

No hay pieza en la
P6.
F207_208_GAP_P6

Orientación correcta
en la P5.
F207_208_ORI_OK_P5

Orientación correcta
en la P6.
F207_208_ORI_OK_P6

Orientación correcta
en la P5.
F207_208_ORI_OK_P5

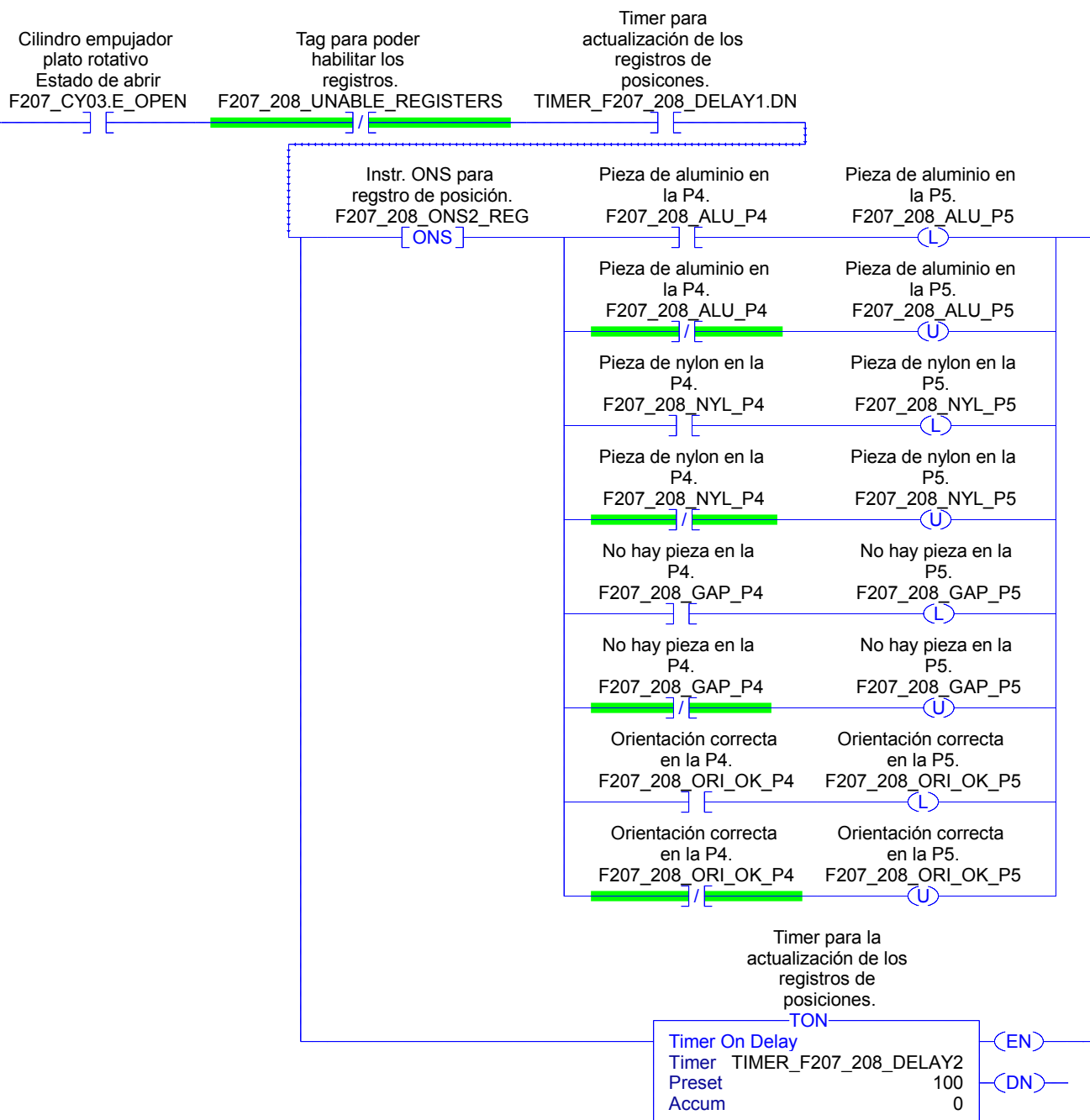
Orientación correcta
en la P6.
F207_208_ORI_OK_P6

Timer para
actualización de los
registros de
posicones.

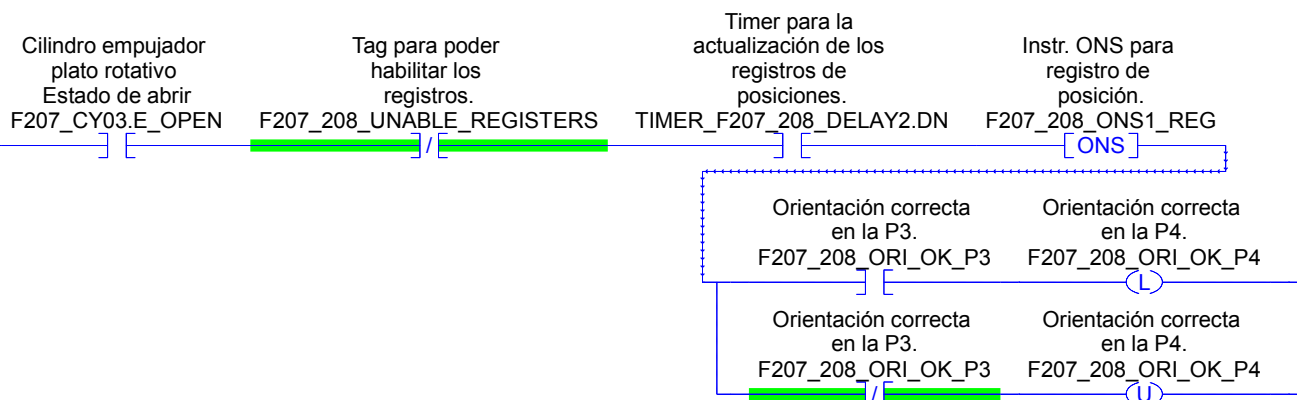
TON
Timer On Delay
Timer TIMER_F207_208_DELAY1
Preset 100
Accum 0

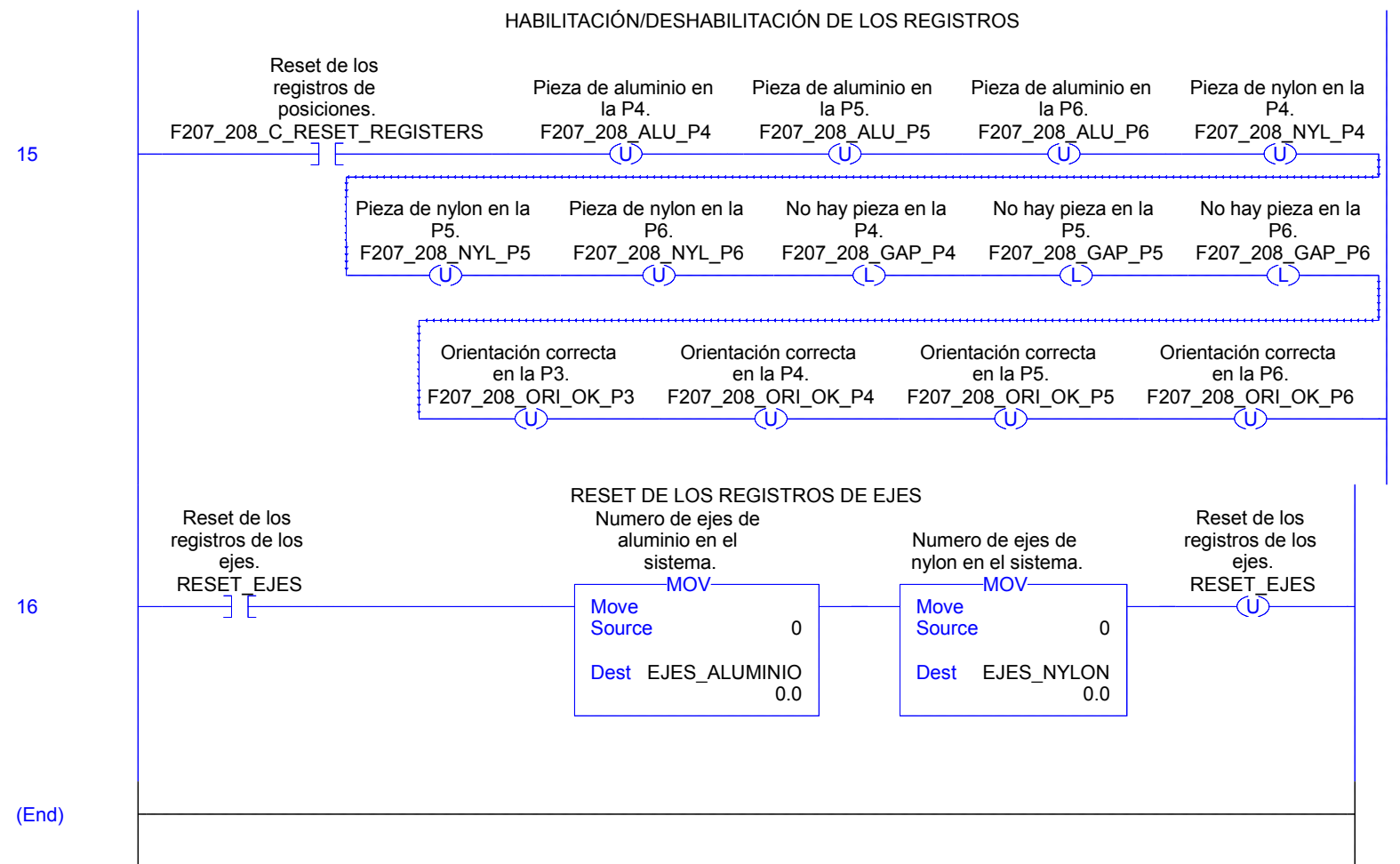
(EN)
(DN)

ACTUALIZACIÓN REGISTROS P4 A P5 (POSICIONES)

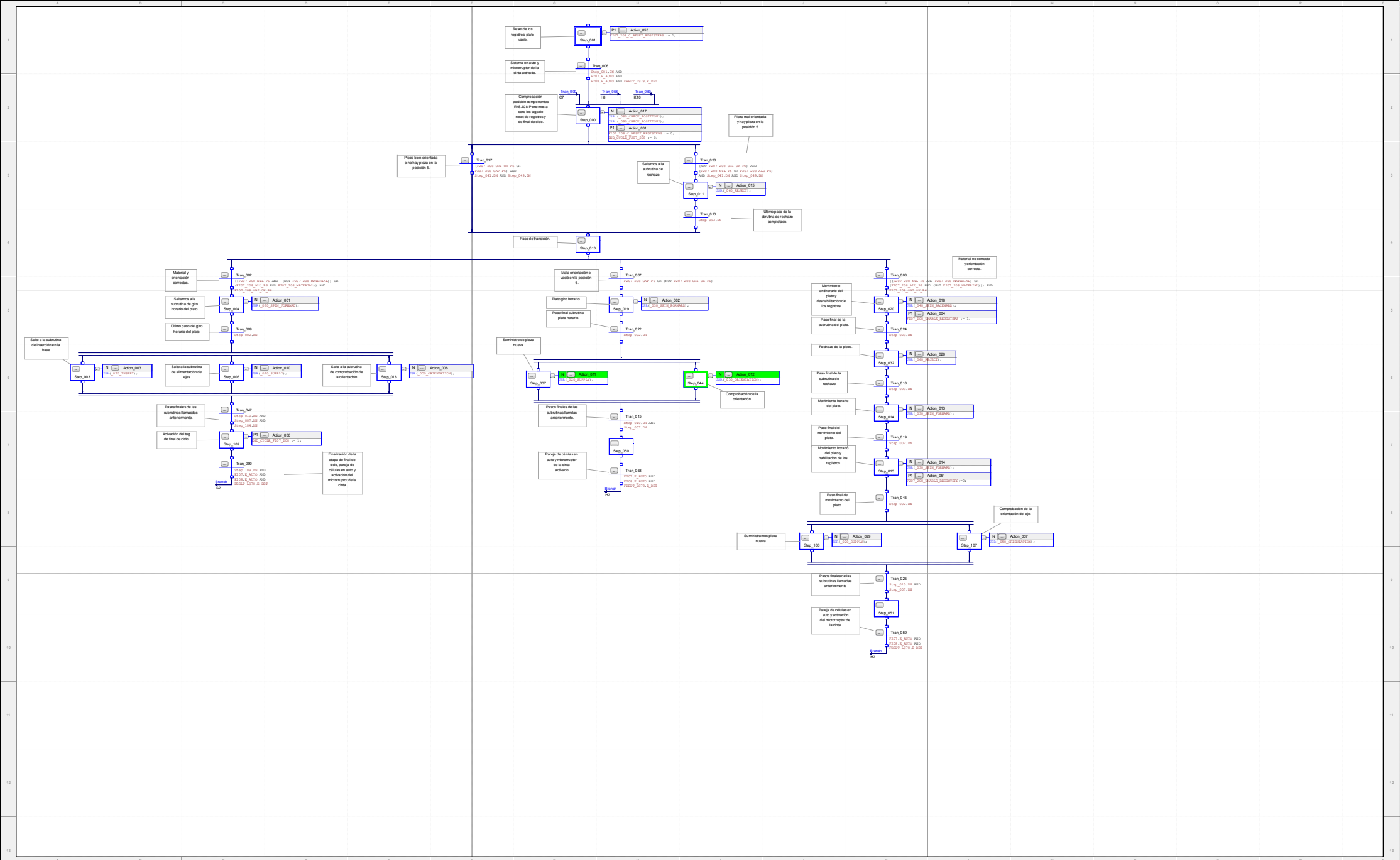


ACTUALIZACIÓN REGISTROS P3 A P4 (POSICIONES)

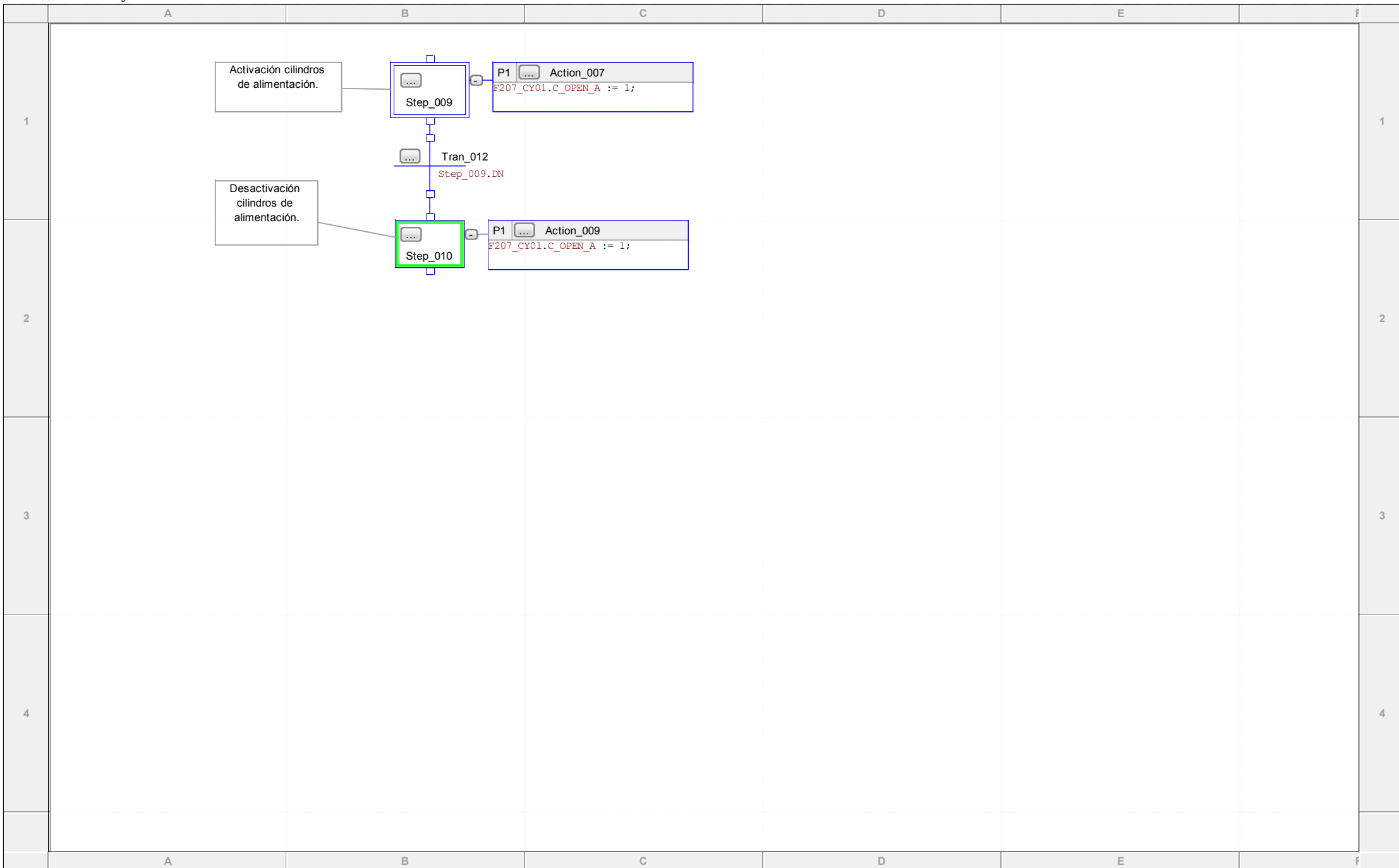




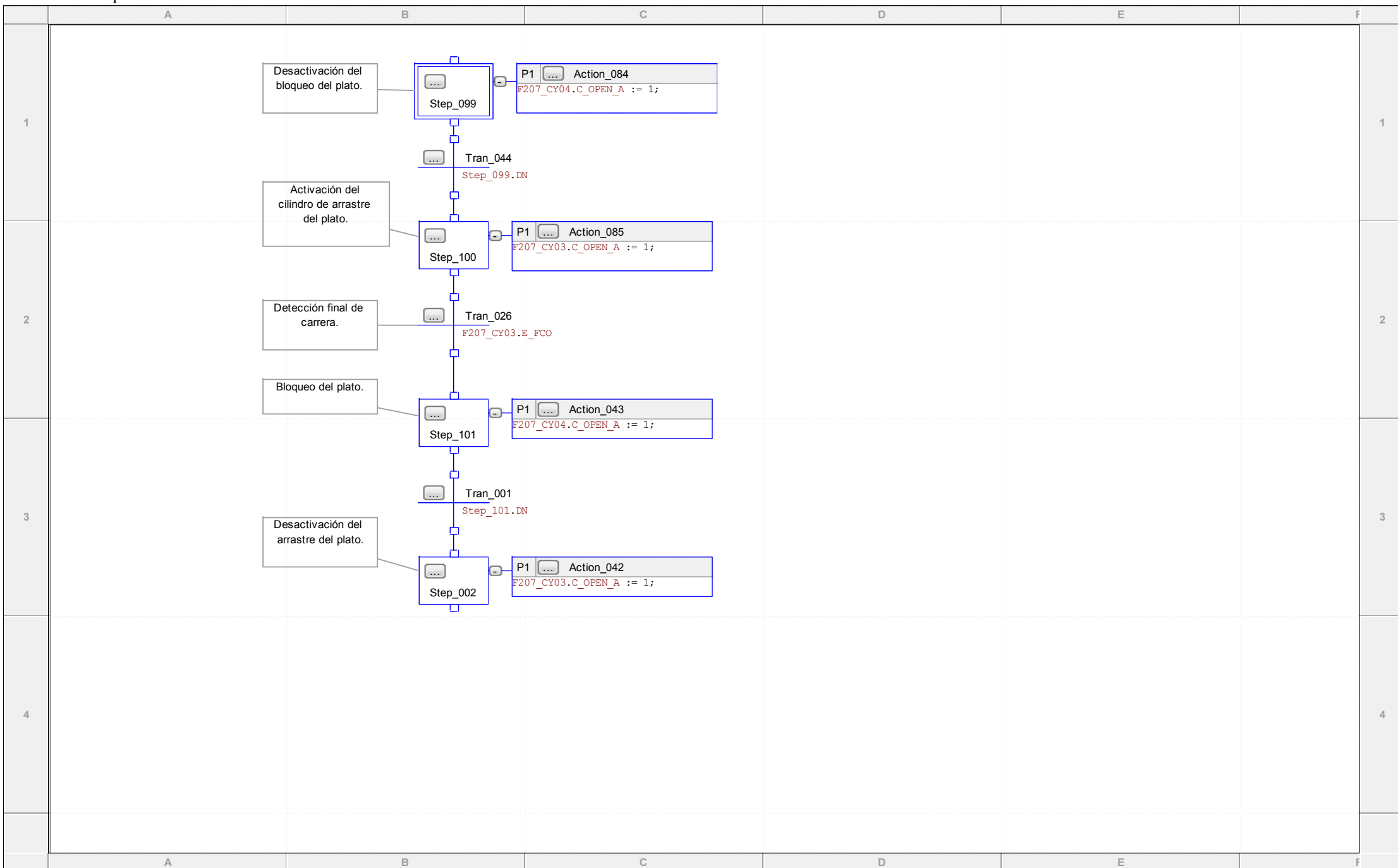
Comportamiento de la FAS 207 y 208.



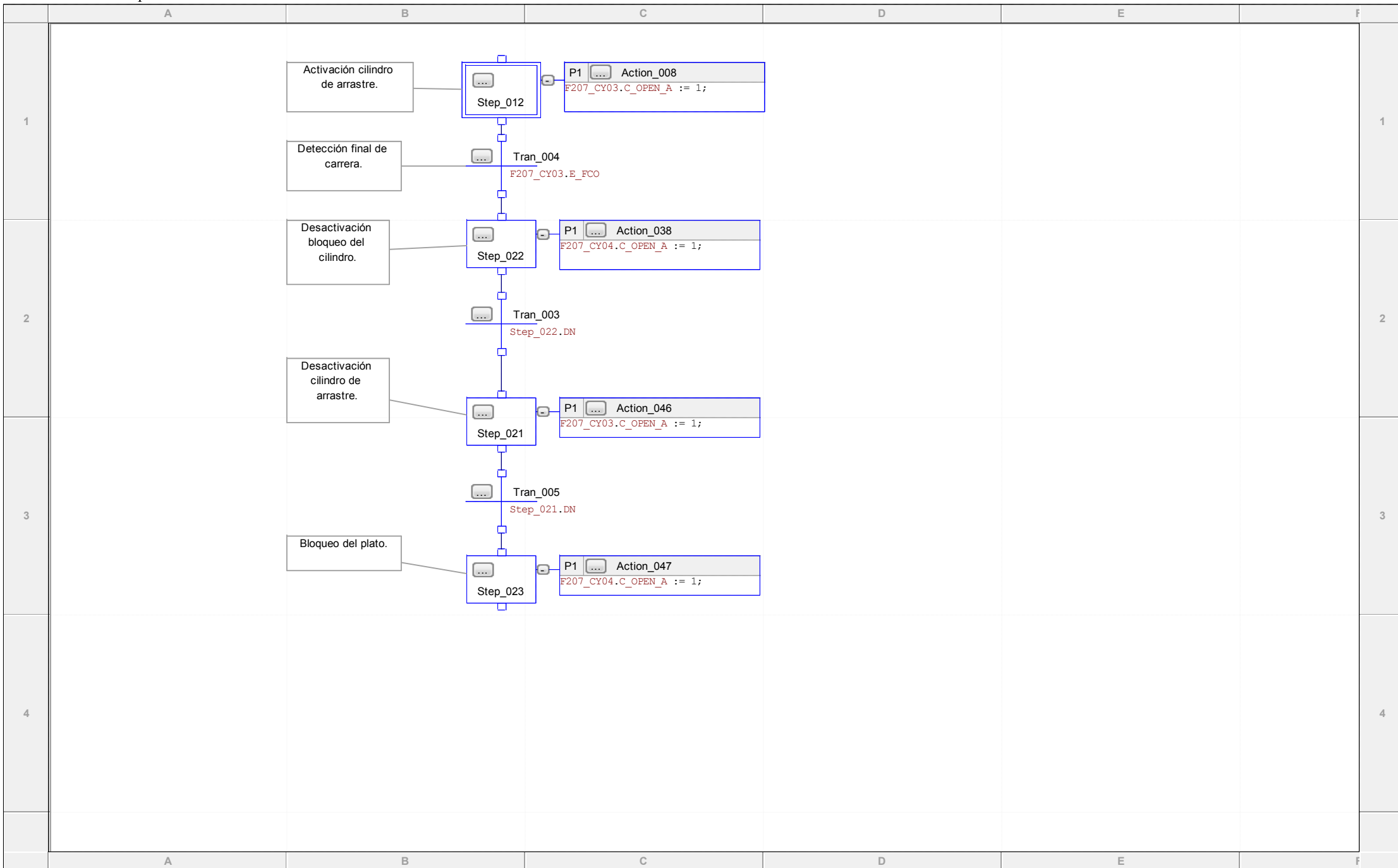
Alimentación de ejes al sistema.



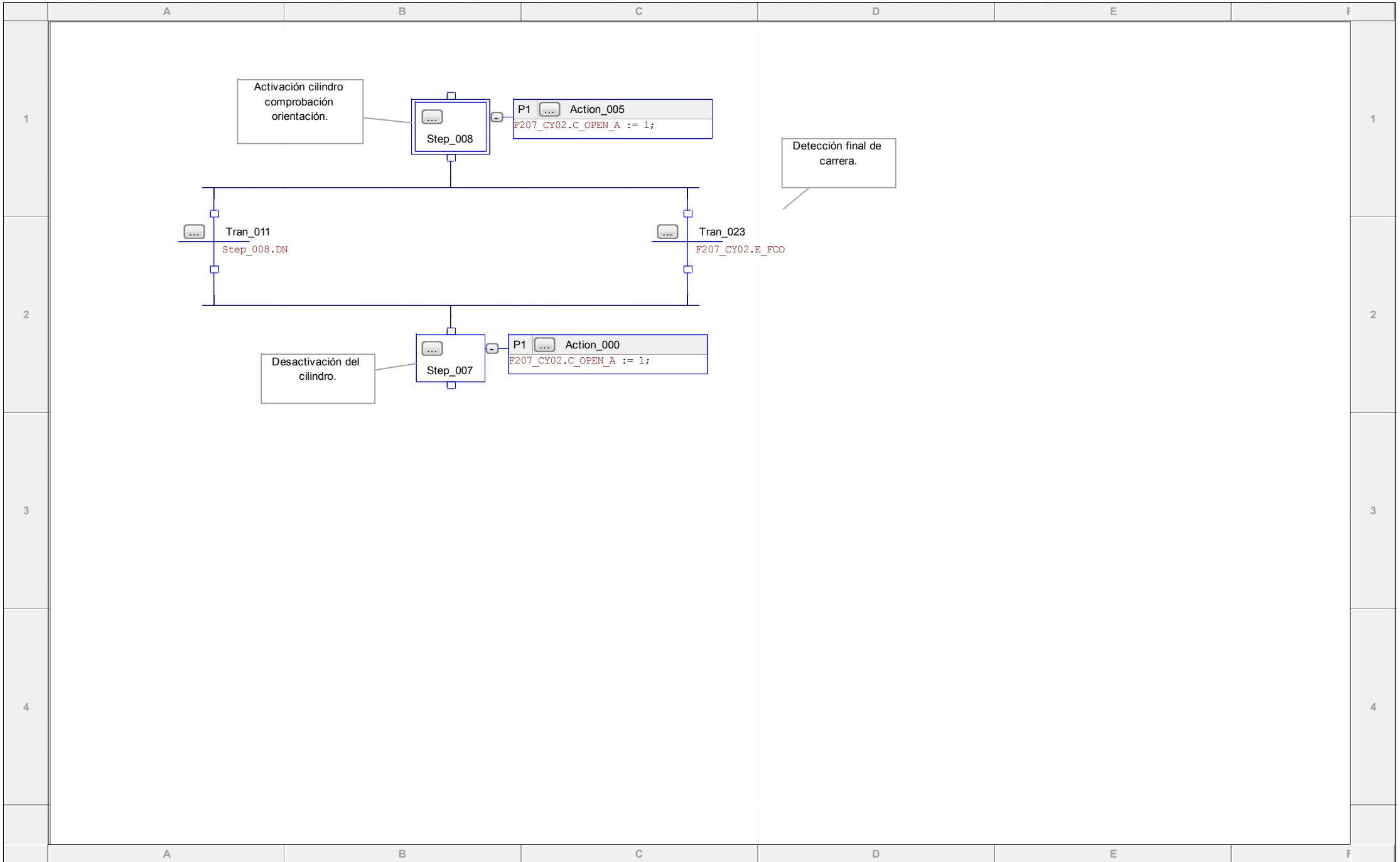
Giro horario del plato.



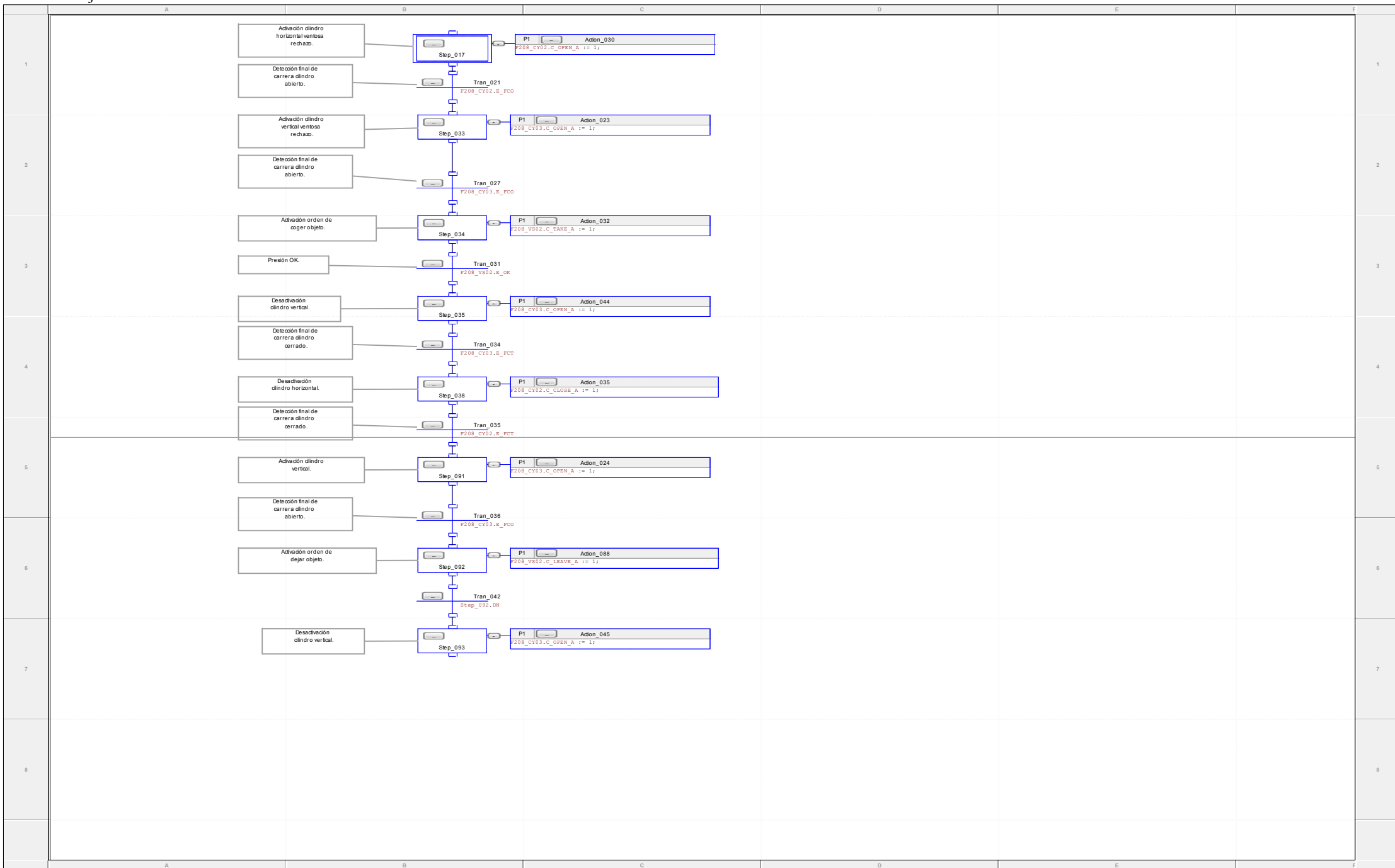
Giro antihorario del plato.



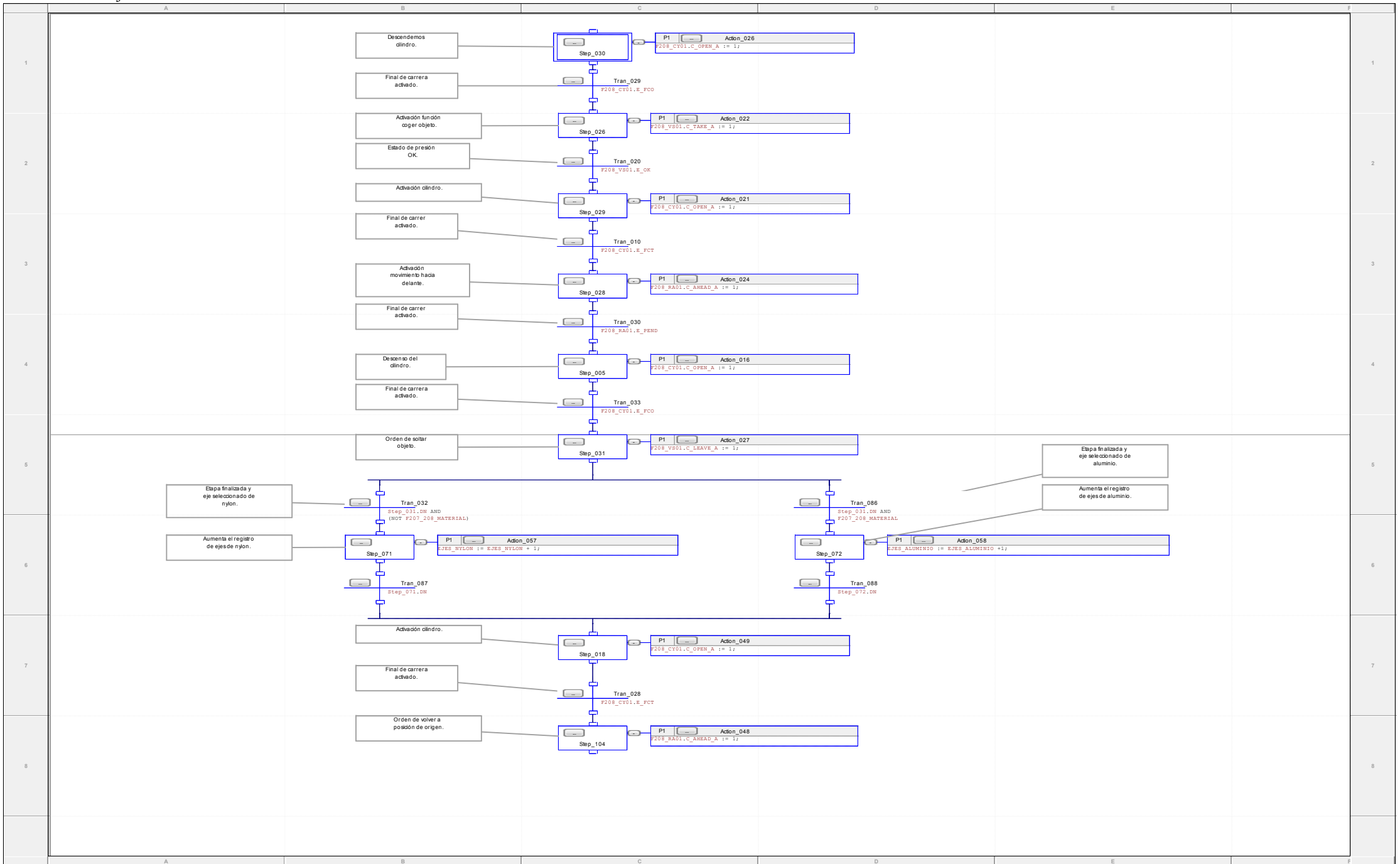
Comprobación de la orientación de los ejes.



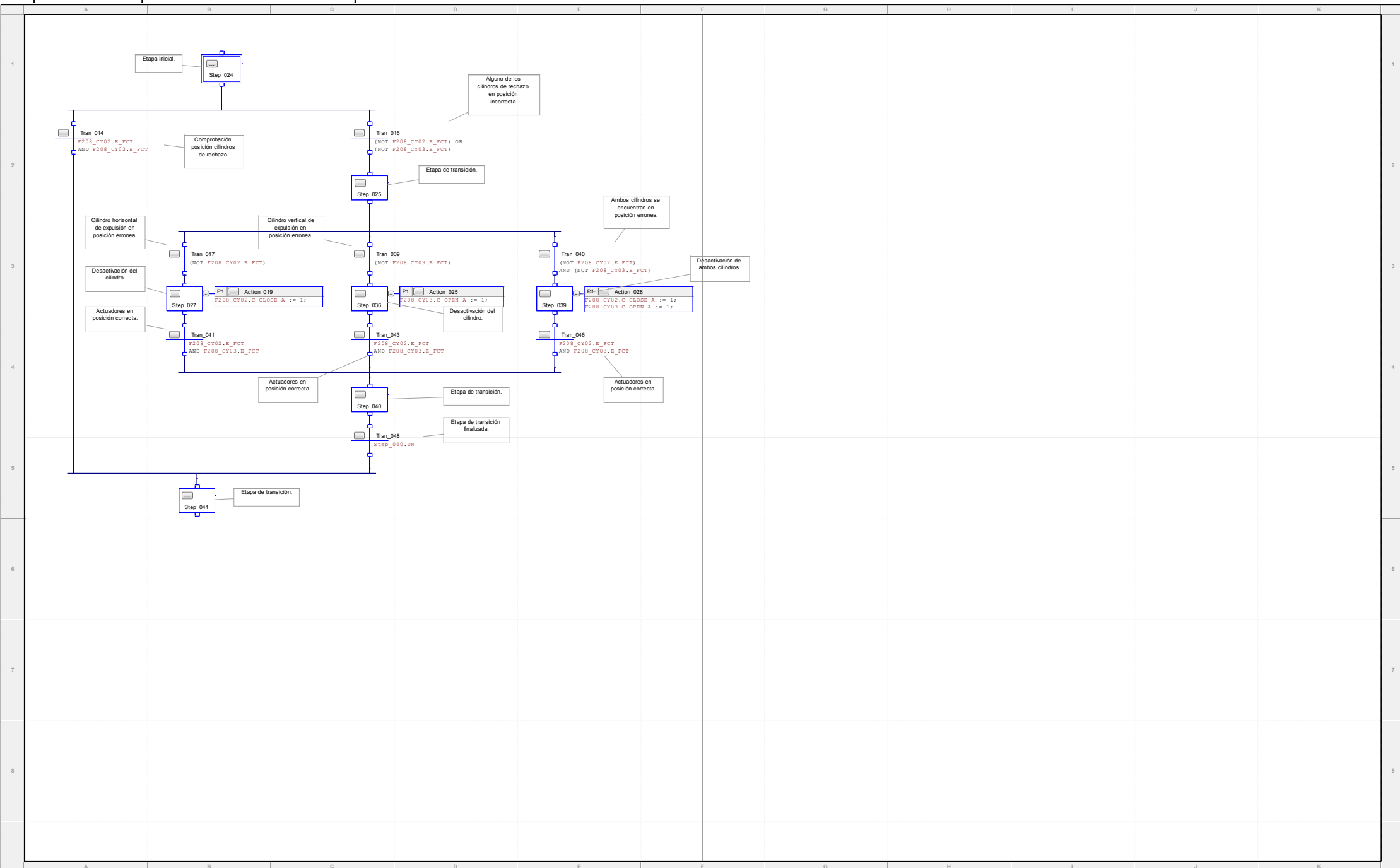
Rechazo de ejes no deseados.



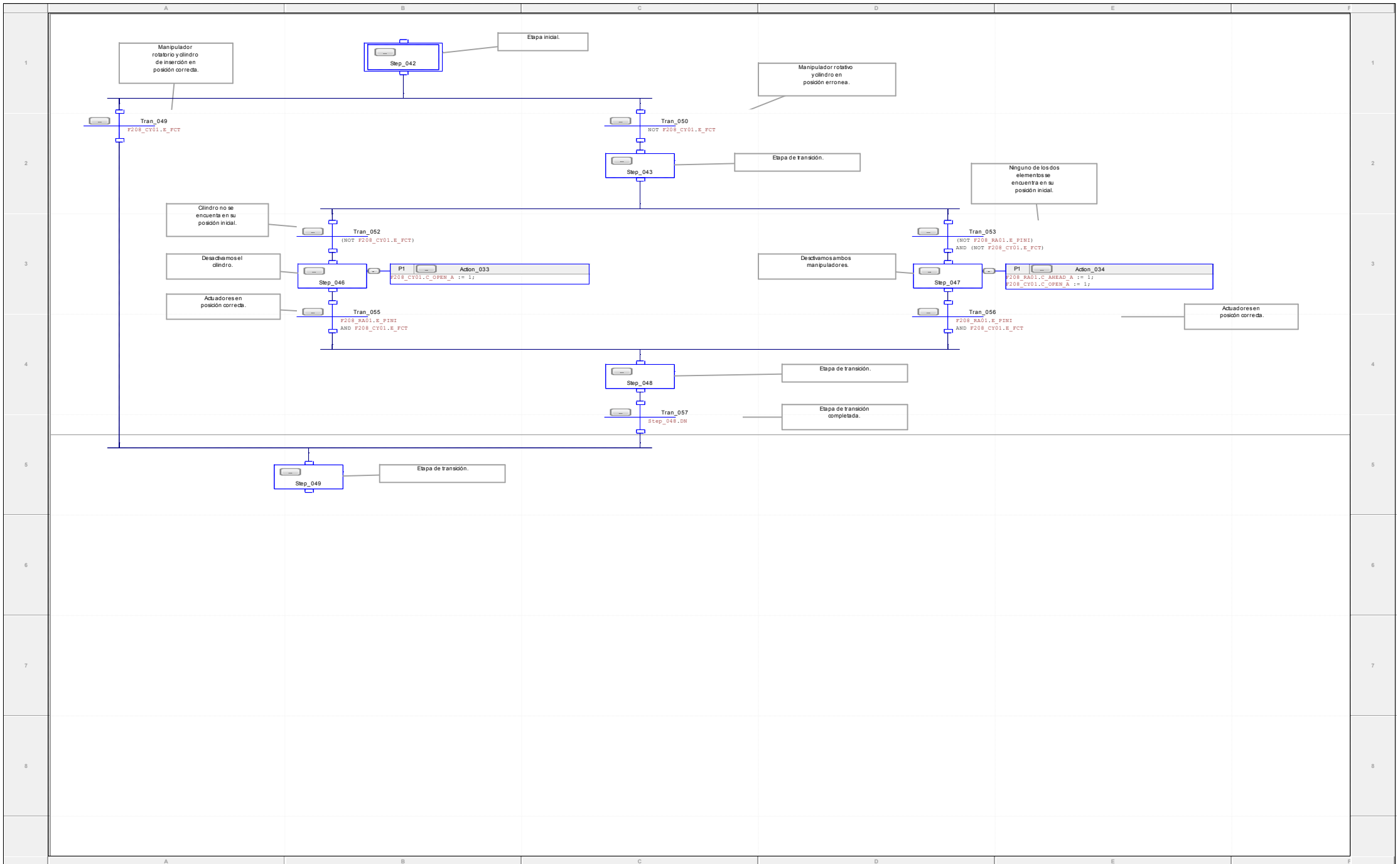
Inserción de los ejes en la base.



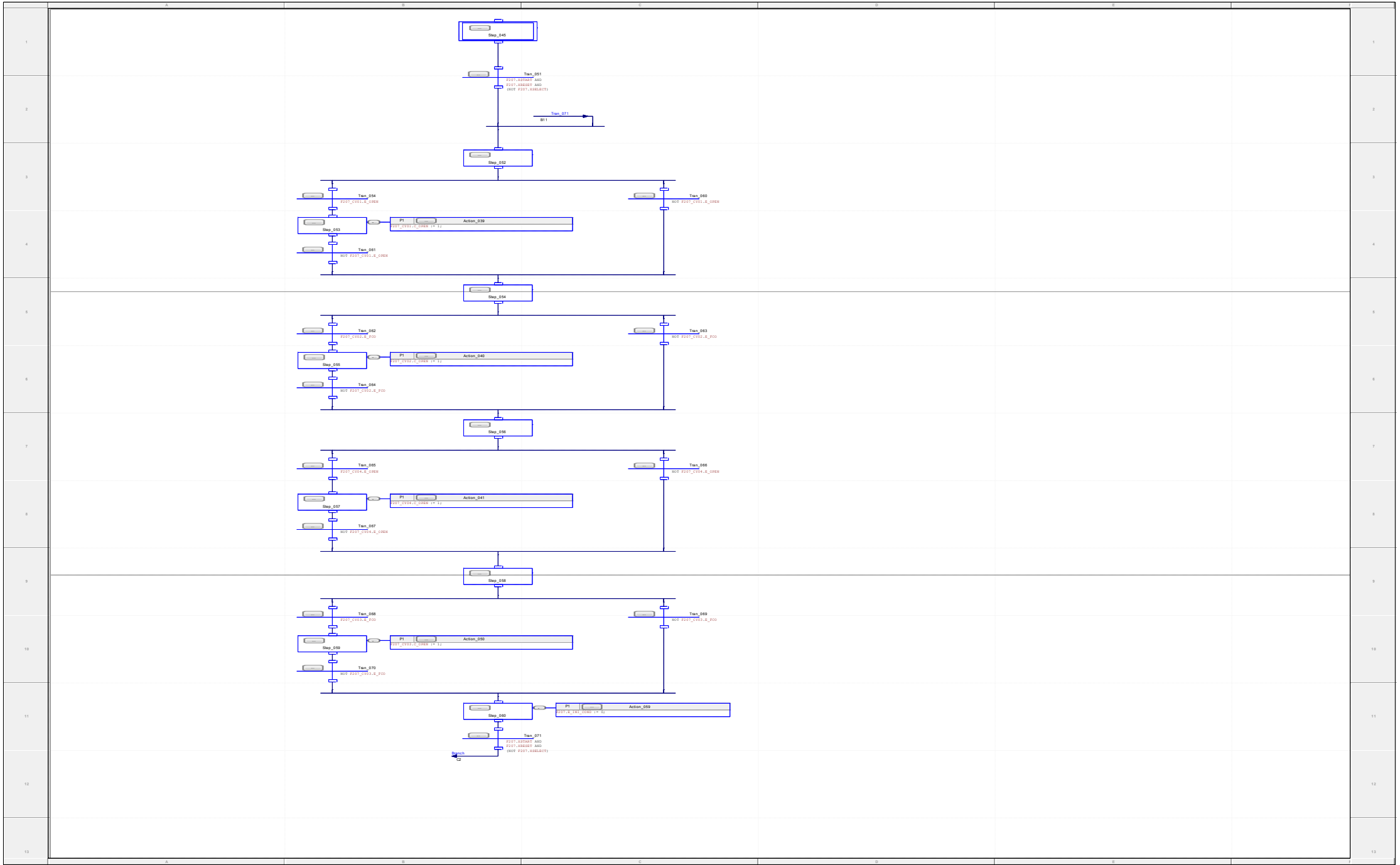
Comprobación de la posición de los elementos de explosión.



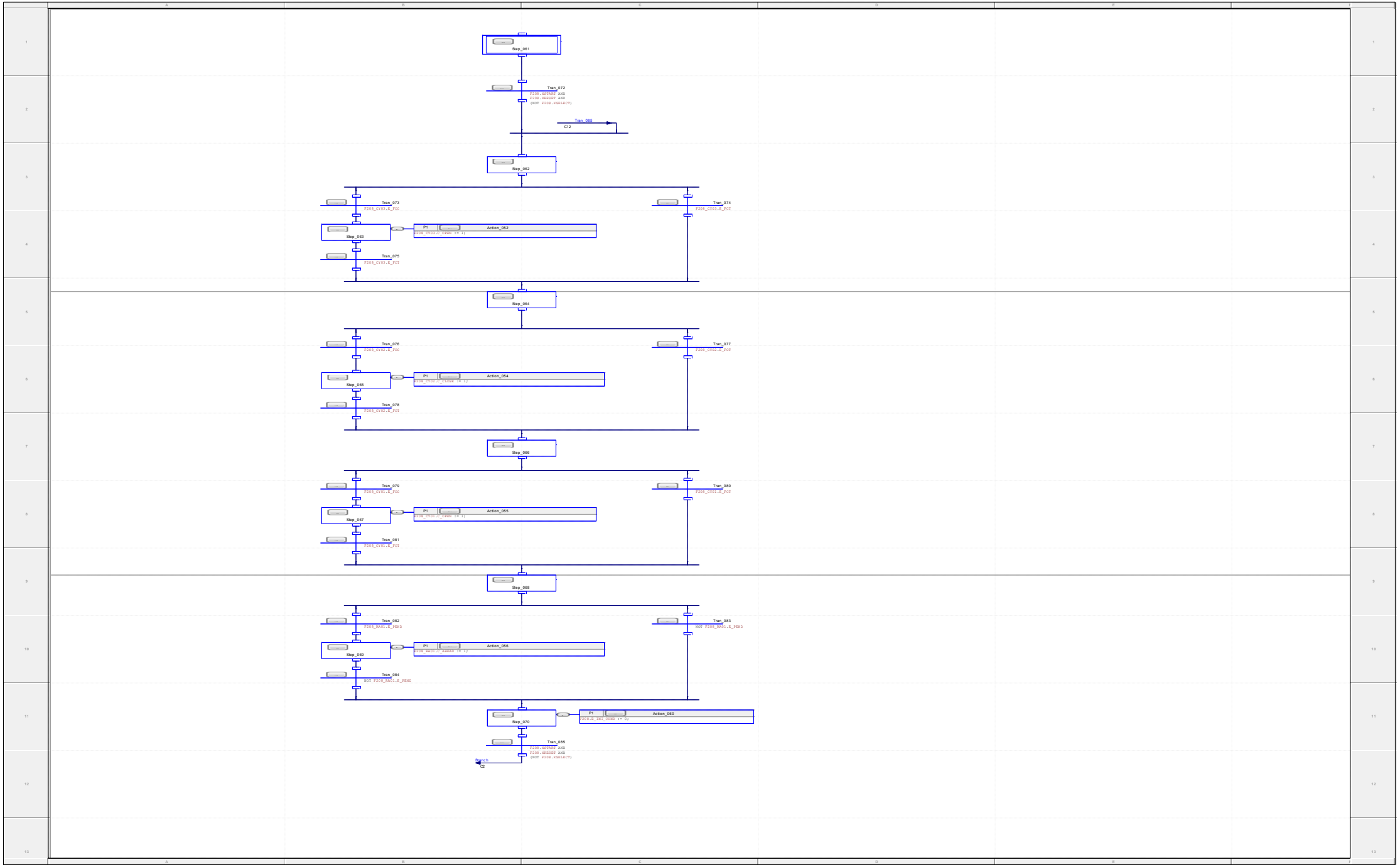
Posicionamiento de los elementos de inserción.



Condiciones iniciales de los actuadores de la FAS207.



Condiciones iniciales de los actuadores de la FAS208.



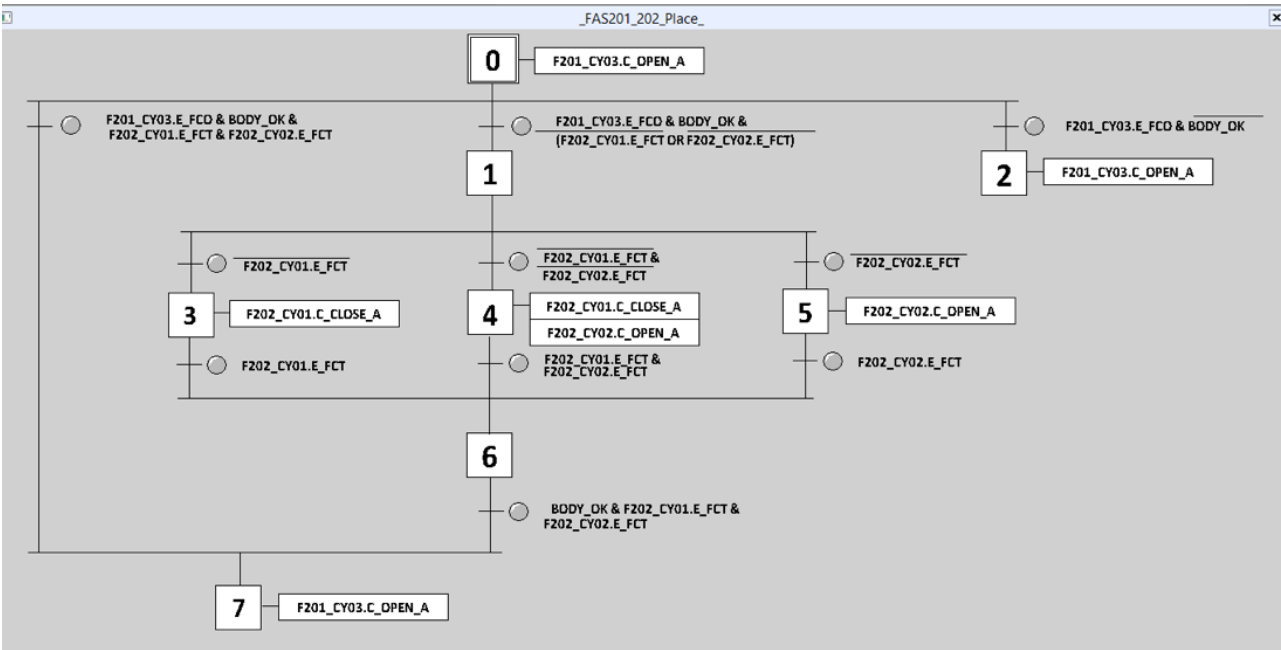
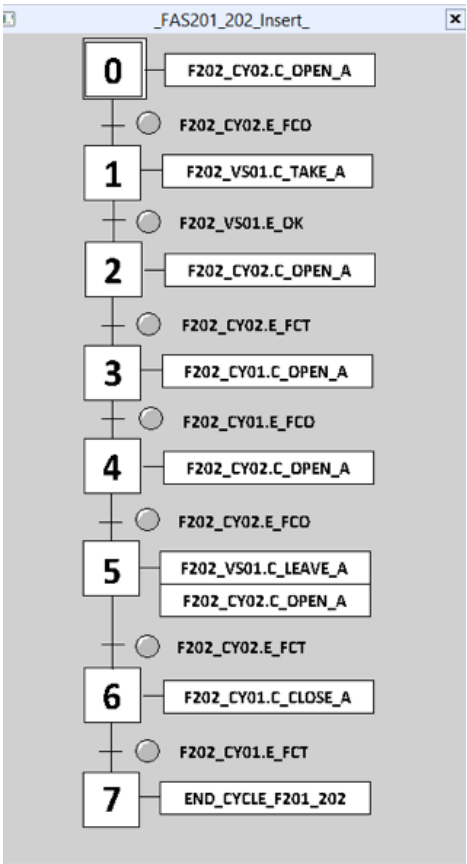
ANEXO 2

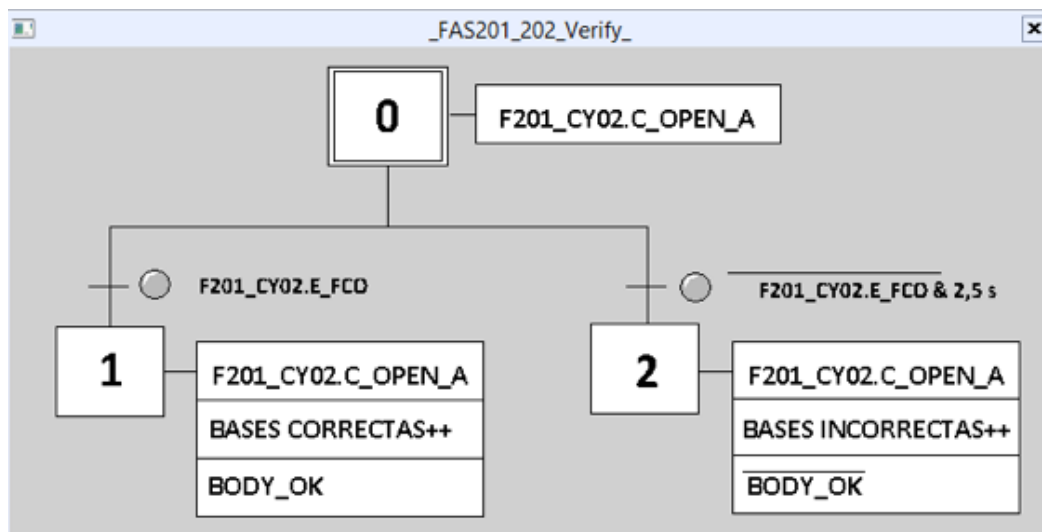
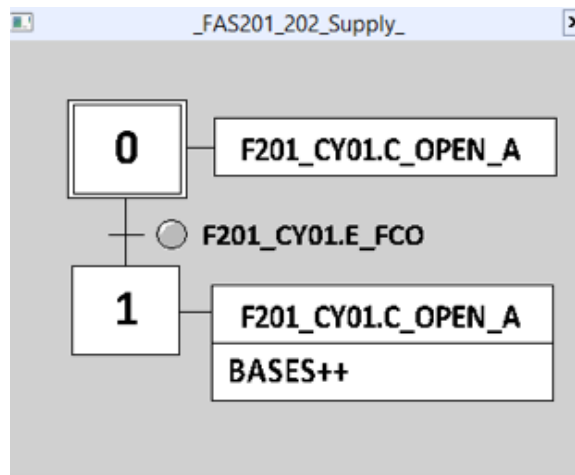
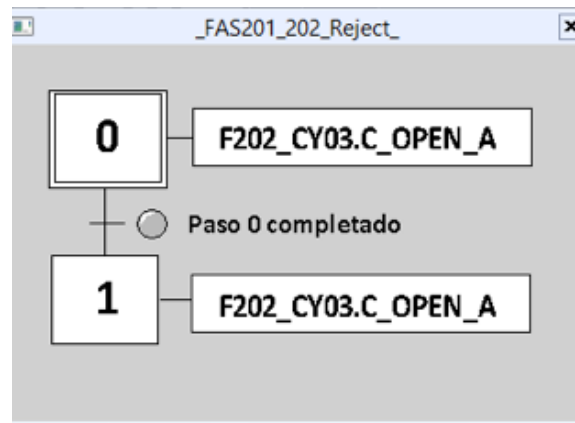
PANTALLAS DE LA APP SCADA

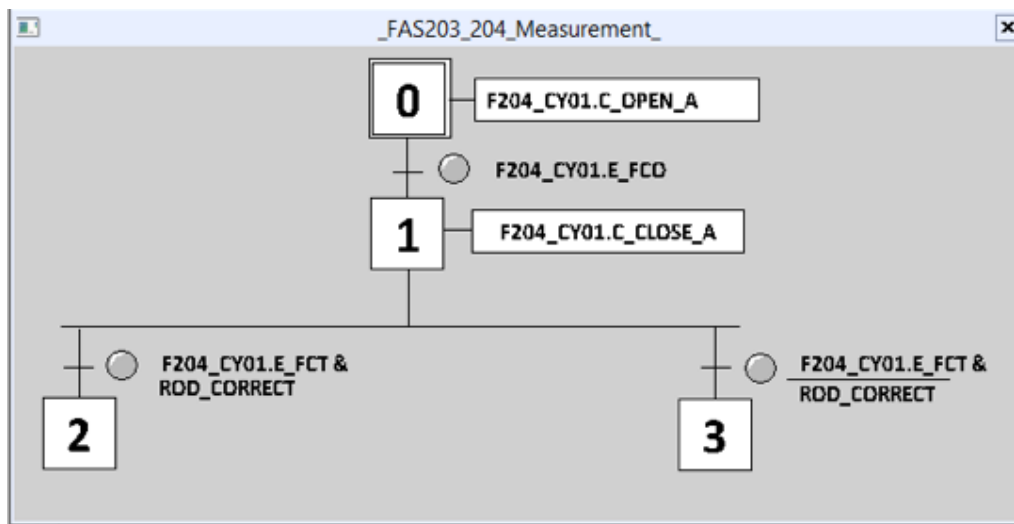
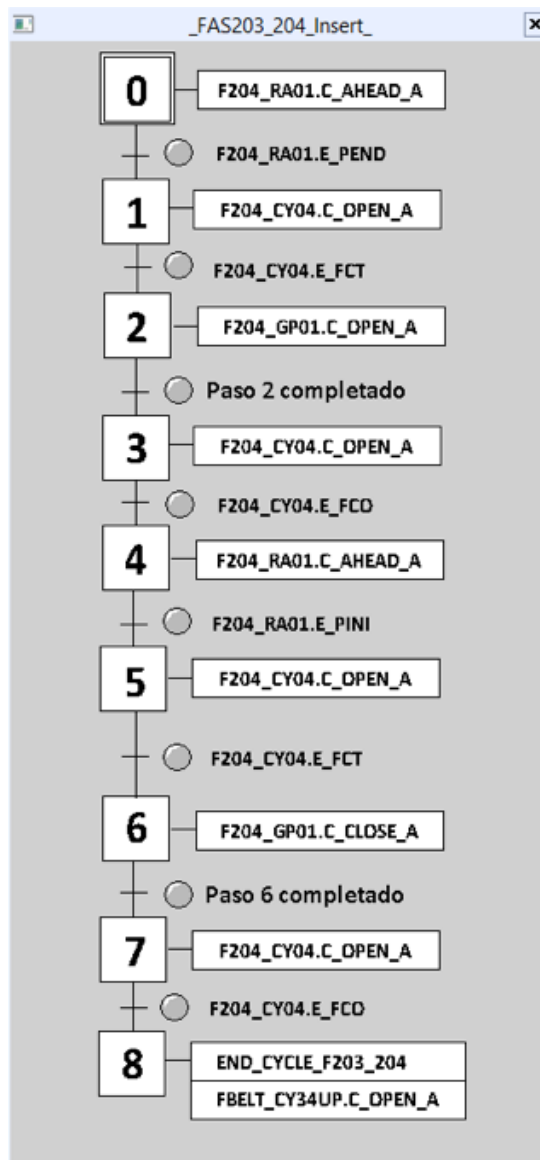
Supervisión Control y Adquisición de Datos

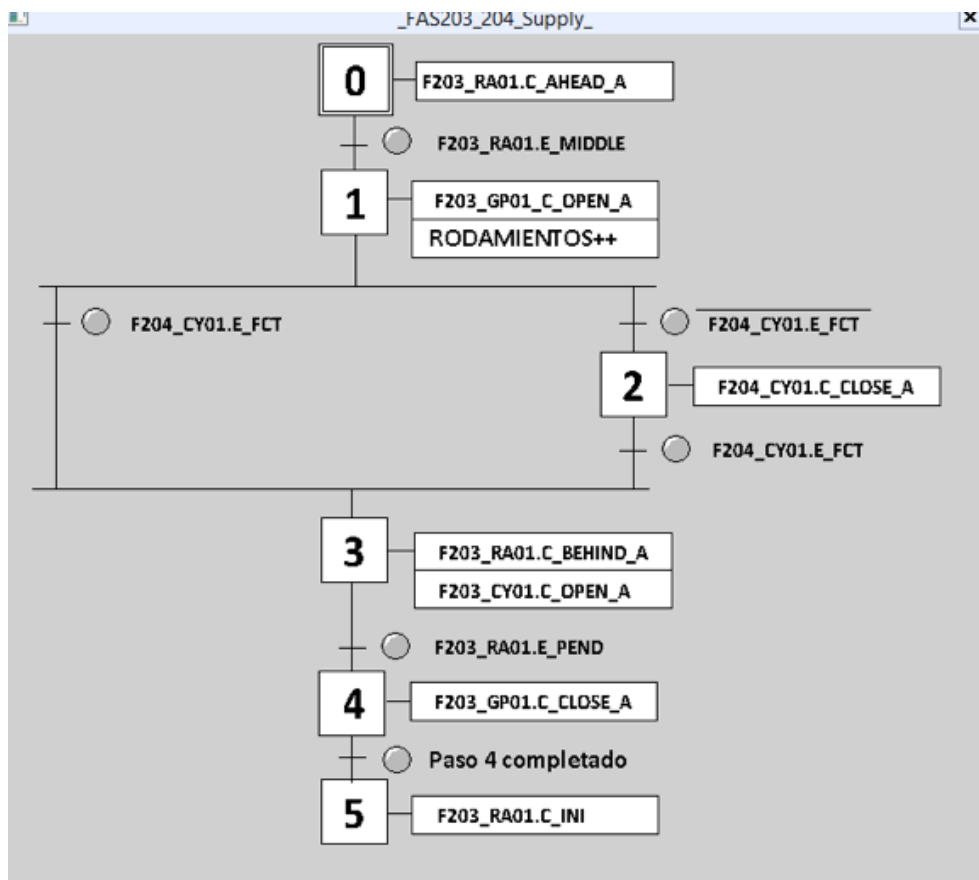
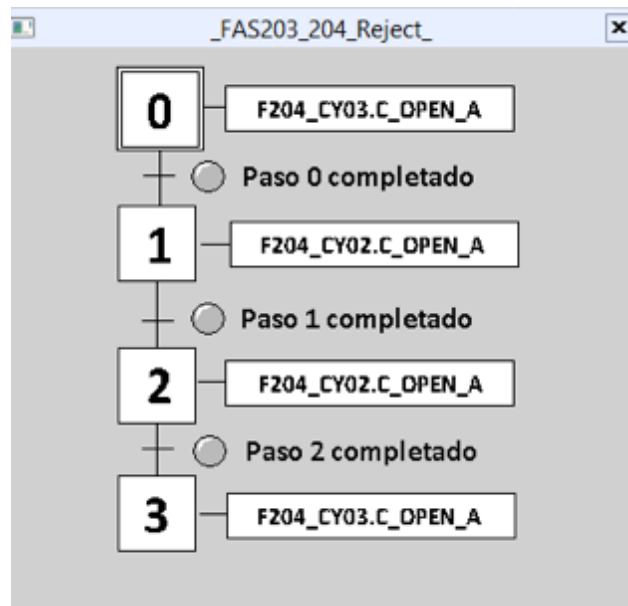
En este segundo anexo se muestran las ventanas de elementos, procesos y modos de funcionamiento que no se han mostrado anteriormente en el proyecto.

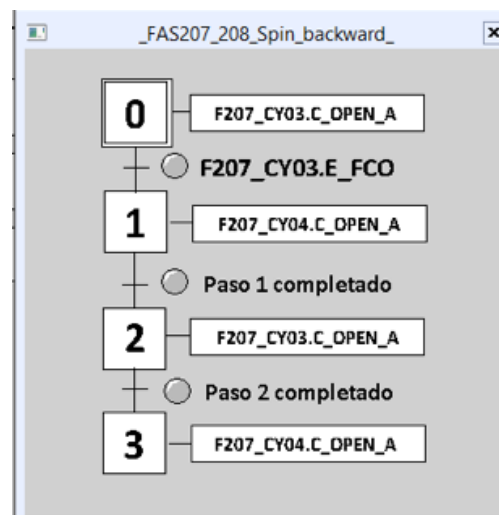
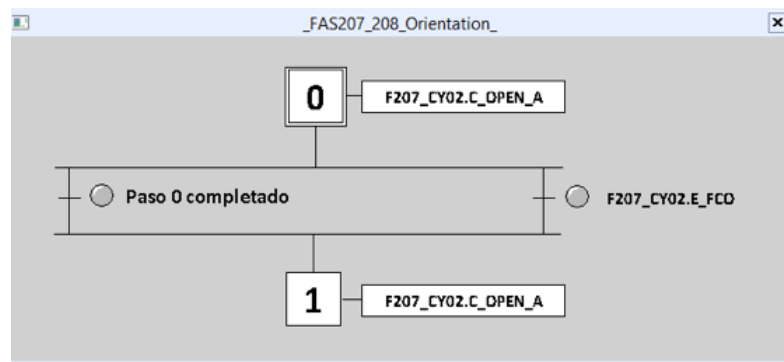
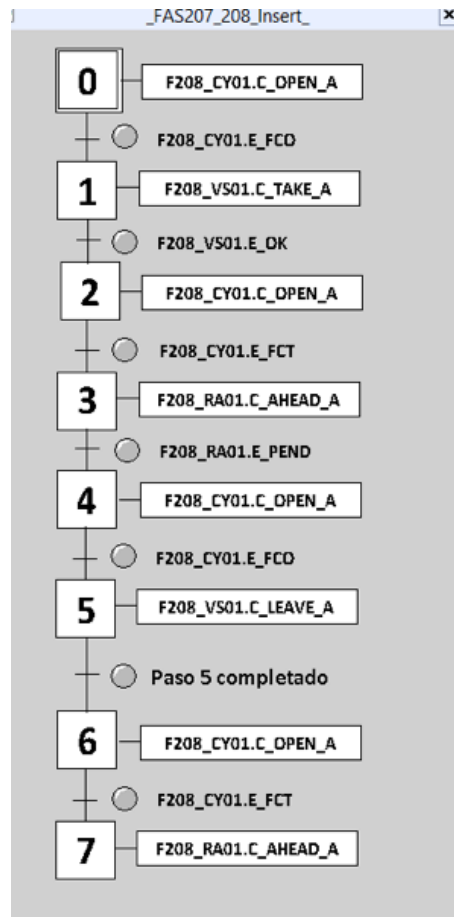
Ventanas de proceso

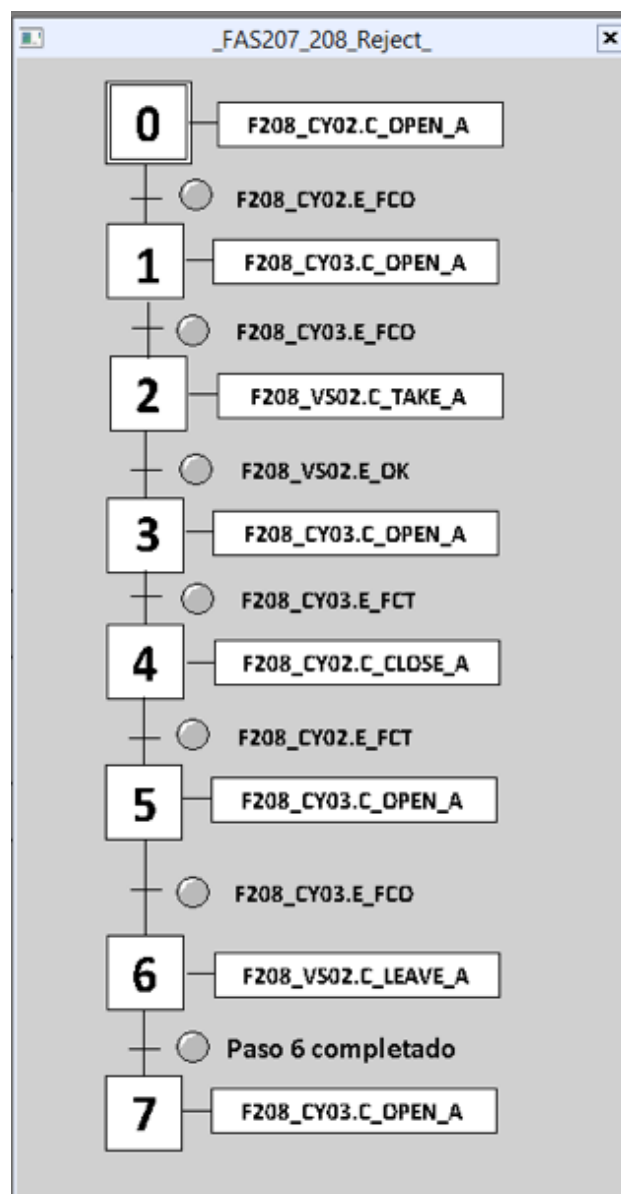
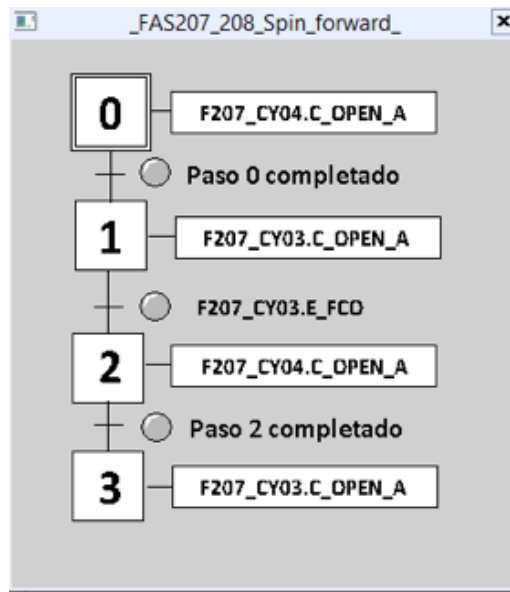


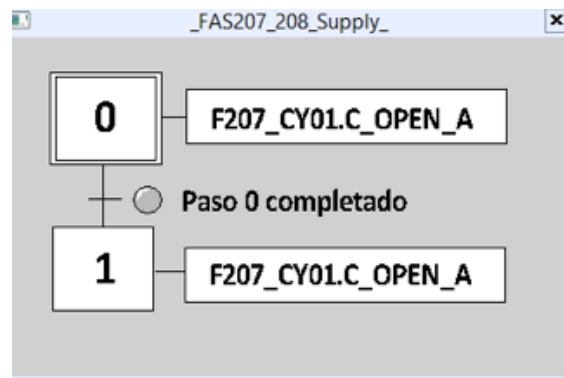




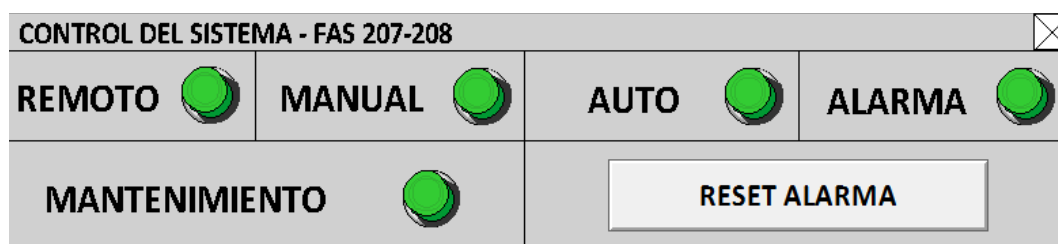
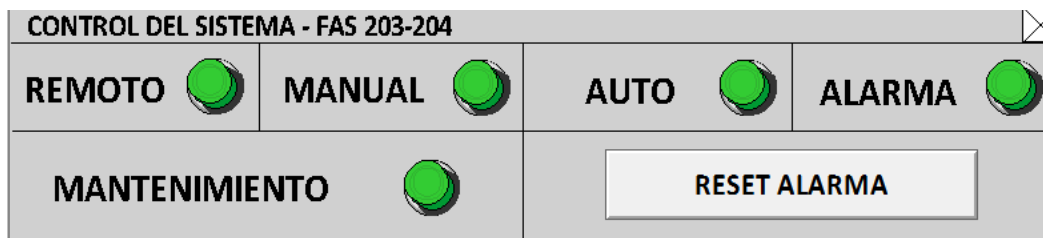








Ventanas de modos de funcionamiento



Ventanas de elementos del sistema

_FAS201_2_Belt_

ELEMENTO	ACCIÓN	SENSORES	ALARMA
FBELT_CY12	OPEN	El actual elemento no dispone de sensores.	El actual elemento no dispone de sensores.

FBELT_CY12 - Cilindro elevador del stopper. (FAS201/2).

_FAS201_Place_

ELEMENTO	ACCIÓN	SENSORES	ALARMA
F201_CY03	OPEN	FCO	RESET

F201_CY03 - Cilindro posicionador en la siguiente célula. (FAS201).

_FAS201_Supply_











ELEMENTO	ACCIÓN	SENSORES	ALARMA
F201_CY01	OPEN	FCO FCT	RESET

F201_CY01 - Cilindro alimentador de bases. (FAS201).



_FAS201_Verify_

ELEMENTO	ACCIÓN	SENSORES	ALARMA
F201_CY02	OPEN	FCO	RESET




F201_CY02 - Cilindro verificador orientación de las bases. (FAS201).

ELEMENTO	ACCIÓN	SENSORES		ALARMA
F202_CY01	OPEN  CLOSE 	FCO 	FCT 	RESET
F202_CY02	OPEN 	FCO 	FCT 	RESET
F202_VS01	TAKE  LEAVE 	E_OK 		RESET










F202_CY01 - Cilindro horizontal de inserción de la base. (FAS202).
 F202_CY02 - Cilindro vertical de inserción de la base. (FAS202).
 F202_VS01 - Ventosas de enganche del manipulador. (FAS202).

ELEMENTO	ACCIÓN	SENSORES	ALARMA 
F202_CY03	OPEN 	El actual elemento no dispone de sensores.	El actual elemento no dispone de sensores.








F202_CY03 - Cilindro rechazador de bases. (FAS202).

ELEMENTO	ACCIÓN	SENSORES	ALARMA 
FBELT_CY34	OPEN 	El actual elemento no dispone de sensores.	El actual elemento no dispone de sensores.
FBELT_CY34UP	OPEN 	El actual elemento no dispone de sensores.	El actual elemento no dispone de sensores.






FBELT_CY34 - Cilindro elevador del stopper. (FAS203/4).
 FBELT_CY34UP - Cilindro elevador del palet. (FAS203/4).

ELEMENTO	ACCIÓN	SENSORES			ALARMA
F203_RA01	AHEAD  BEHIND  PINI	PINI 	MIDDLE 	PEND 	RESET
F203_CY01	OPEN 	FCO 			RESET
F203_GP01	OPEN  CLOSE 	El actual elemento no dispone de sensores.			El actual elemento no dispone de sensores.




F203_RA01 - Actuador rotativo alimentador de rodamientos. (FAS203).
 F203_CY01 - Cilindro alimentador de rodamiento al sistema. (FAS203).
 F203_GP01 - Pinza neumática que sujeta el rodamiento en todo el trayecto efectuado.

ELEMENTO	ACCIÓN	SENSORES	ALARMA
F204_CY04	OPEN 	FCO 	RESET
F204_RA01	AHEAD 	PINI  PEND 	RESET
F204_GP01	OPEN  CLOSE 	El actual elemento no dispone de sensores.	




F204_CY04 - Cilindro vertical del manipulador rotatorio de inserción de rodamiento. (FAS204).
 F204_RA01 - Manipulador rotativo para insertar rodamientos en la base. (FAS204).
 F204_GP01 - Pinzas neumáticas para trasladar el rodamiento.

ELEMENTO	ACCIÓN	SENSORES	ALARMA 
F204_CY01	OPEN  CLOSE 	FCO  FCT 	RESET




F204_CY01 - Cilindro de la medición de altura del rodamiento. (FAS204).



ELEMENTO	ACCIÓN	SENSORES	ALARMA 
F204_CY02	OPEN 	El actual elemento no dispone de sensores.	El actual elemento no dispone de sensores.
F204_CY03	OPEN 	El actual elemento no dispone de sensores.	El actual elemento no dispone de sensores.










F204_CY02 - Cilindro de expulsión de rodamientos erróneos. (FAS204).
 F204_CY03 - Cilindro de centrado del rodamiento en su posición de medición.











ELEMENTO	ACCIÓN	SENSORES	ALARMA 
F207_CY02	OPEN 	FCO 	RESET



F207_CY02 - Cilindro verificador de la orientación de los ejes. (FAS207).

_FAS207_Spin_			
ELEMENTO	ACCIÓN	SENSORES	ALARMA
F207_CY03	OPEN 	FCO 	RESET
F207_CY04	OPEN 	El actual elemento no dispone de sensores.	El actual elemento no dispone de sensores.
F207_CY03 - Cilindro de movimiento del plato rotatorio. (FAS207). F207_CY04 - Cilindro de bloqueo del plato giratorio. (FAS207).			

_FAS207_Supply_			
ELEMENTO	ACCIÓN	SENSORES	ALARMA 
F207_CY01	OPEN 	El actual elemento no dispone de sensores.	El actual elemento no dispone de sensores.
F207_CY01 - Cilindro alimentador de ejes. (FAS207).			

_FAS208_Insert_				
ELEMENTO	ACCIÓN		SENSORES	ALARMA
F208_RA01	AHEAD 		PINI  PEND 	RESET
F208_CY01	OPEN 		FCO  FCT 	RESET
F208_VS01	TAKE  LEAVE 		E_OK 	RESET
F208_RA01 - Actuador rotativo de inserción del eje en la base. (FAS208). F208_CY01 - Cilindro vertical de inserción del eje en la base. (FAS208). F208_VS01 - Ventosa de inserción del eje en la base. (FAS208).				

_FAS208_Reject_				
ELEMENTO	ACCIÓN		SENSORES	ALARMA
F208_CY02	OPEN  CLOSE 		FCO  FCT 	RESET
F208_CY03	OPEN 		FCO  FCT 	RESET
F208_VS02	TAKE  LEAVE 		E_OK 	RESET
F208_CY02 - Cilindro horizontal de expulsión de ejes incorrectos. (FAS208). F208_CY03 - Cilindro vertical de expulsión de ejes incorrectos. (FAS208). F208_VS02 - Ventosa de expulsión de ejes incorrectos. (FAS208).				

_FAS207_8_Belt_ ✕			
ELEMENTO	ACCIÓN	SENSORES	ALARMA 
FBELT_CY78	<div>OPEN </div>	El actual elemento no dispone de sensores.	El actual elemento no dispone de sensores.
FBELT_CY78 - Cilindro elevador del stopper. (FAS207/8).			

ANEXO 3

MEMORIA ECONÓMICA

En el actual anexo se presenta la memoria económica. Se evalúan los costes de mano de obra de una puesta en marcha de estas características. Suponiendo, desde un principio, que se dispone de todo el *hardware* y no debemos tenerlo en cuenta.

Concepte		Horas Dedicadas (h)	Precio Unitario (€/h)	Total (€)
Análisis	Análisis FAS 200	25 h	15 €/h	375 €
Diseño	Codificación de elementos y sistemas	25 h	15 €/h	375 €
	Programación PLC	250 h	15 €/h	3.000 €
	Diseño SCADA	125 h	15 €/h	1.875 €
Implementación	Pruebas FAT	50h	15 €/h	750 €
	Pruebas SAT	75 h	25 €/h	1.875 €
	Documentación	120 h	15 €/h	1.800 €
Total sin IVA				10.050 €
IVA 21%				
TOTAL				12.160 €